



NORTHWESTERN UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2026-1

March, 2026

Efficient Vision-based Methods for Hand-related Activity Recognition

Soroush Shahi

Abstract

Vision-based sensing is central to wearable and spatial computing because it enables recognition of hand-related activities important for health monitoring and human–computer interaction, yet continuous RGB video capture is power-intensive, data-heavy, computationally demanding, and raises privacy concerns. This document addresses these challenges by introducing efficient methods for hand activity recognition that reduce power consumption, data capture, and training requirements with minimal impact on accuracy. It demonstrates that low-resolution thermal sensors can serve as a power-efficient alternative or complement to RGB cameras, fully supporting recognition of activities with distinct thermal signatures, such as smoking, and improving performance in more complex cases like eating. Building on this multimodal insight, the work proposes an adaptive sampling framework in which thermal cues guide spatiotemporal RGB capture, substantially reducing video volume and energy use while preserving high recognition accuracy. Finally, the dissertation introduces a meta-learning-based framework that enables personalized hand gesture recognition from a single visual demonstration, reducing user burden and training overhead. Together, these contributions advance practical, efficient, and privacy-aware vision-based wearable systems for hand activity recognition.

Human Activity Recognition, Computer Vision, Wearable Devices

NORTHWESTERN UNIVERSITY

Efficient Vision-based Methods for Hand-related Activity Recognition

A DISSERTATION
SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Soroush Shahi

EVANSTON, ILLINOIS

March 2026

ABSTRACT

Vision-based sensing has become increasingly central to wearable and spatial computing devices because of its ability to capture rich contextual cues about hand-related activities. These activities, ranging from everyday behaviors such as eating and smoking to intentional interactions such as hand gestures, are critical both for health monitoring and for human-computer interaction. However, the reliance on continuous visual data creates fundamental challenges. Continuous RGB video capture consumes significant power, generates excessive amounts of data, raises privacy concerns, and demands substantial computational resources for real-time inference. This dissertation addresses these challenges by introducing methods to improve the efficiency of vision-based hand activity recognition by reducing power consumption, minimizing data capture, and lowering training data requirements with minimal impact on recognition accuracy.

This work first establishes the utility of low-resolution thermal sensors as a power-efficient alternative to traditional cameras. For activities with distinct thermal signatures, such as smoking, thermal data can entirely replace RGB imagery, enabling unobtrusive and accurate monitoring. In more complex scenarios, such as distinguishing eating from other hand-to-mouth gestures, thermal sensors can augment RGB data to significantly improve detection accuracy.

Building on this multi-modal approach, this dissertation introduces an adaptive sampling strategy for activities that necessitate the detail of RGB imagery. By using thermal cues to guide the spatio-temporal capture of visual data, the system processes only the most informative video segments. This method dramatically reduces power consumption and data volume while preserving high recognition accuracy, making long-term, privacy-respecting monitoring practical. Finally, beyond sensing efficiency, this research also tackles the challenge of data-intensive training. A novel framework for hand gesture recognition is presented, leveraging meta-learning to enable robust

customization from a single visual demonstration. This approach minimizes user burden during data collection and lowers the overhead for training personalized interaction models. Together, these contributions demonstrate a path toward vision-based wearable systems that achieve accurate hand-related activity recognition while remaining practical, efficient, and respectful of user privacy.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Nabil Alshurafa, for his unwavering guidance and support throughout my PhD journey. He constantly challenged my perspectives and pushed me to refine my thinking, helping me learn how to identify research questions that truly matter. I am thankful for his mentorship in developing the skills of clear storytelling, rigorous research, and persuasive communication. His encouragement, patience, and belief in my potential have shaped me both as a researcher and as a person.

I am also sincerely thankful to my committee members. I thank Aggelos Katsaggelos for sharing his deep technical expertise and insightful feedback that helped strengthen my work. I am grateful to Bryan Pardo for his mentorship and encouragement during some of the most challenging moments of my PhD, when his guidance gave me clarity and direction. I also thank Kareem Bedri for his valuable advice and mentorship during my research journey at Apple, which provided me with a broader perspective on bridging academia and industry.

I would like to acknowledge my mentors, Sougata Sen, Rawan Alharbi, Yang Gao, Mahdi Pedram, and Josiah Hester, whose advice and support have been valuable to my development. I am also thankful to my collaborators and lab mates, Farzad Shahabi, Glenn Fernandes, Boyang Wei, Tanmeet Butani, Christopher Romano, Bonnie Nolan, Jeb Sumeracki, and Tim Truty, for their contributions and for fostering a collaborative environment.

Finally, I would like to acknowledge the generous support of the National Institutes of Health and the National Science Foundation, whose funding made this research possible.

To my parents, who taught me to think beyond the immediate

To my siblings, whose presence filled life with laughter

And to Sabaa, for being my source of balance, comfort, and love

TABLE OF CONTENTS

Acknowledgments	3
List of Figures	9
List of Tables	14
Chapter 1: Introduction	16
Chapter 2: Low-Power Thermal Cues for Augmenting RGB-Based Visual Sensing . . .	19
2.1 Problem Statement and Motivation	19
2.2 Related Work	23
2.3 System Design	26
2.4 System Processing Framework	29
2.5 Evaluation	36
2.6 Implications for Design: An Assessment of False Positives and False Negatives . .	40
2.7 Discussion and Future Work	44
2.8 Summary of Findings	46

Chapter 3: Energy-Efficient Smoking Detection using Thermal Imagery	47
3.1 Problem Statement and Motivation	47
3.2 Related Work	50
3.3 SmokeMon: System Design, Implementation and Energy Consumption	56
3.4 SmokeMon Framework to Extract Smoking Topography	65
3.5 Experiments and Datasets	69
3.6 Results	75
3.7 Discussion and Future Work	83
3.8 Summary of Findings	86
Chapter 4: Thermal-Guided Visual Sampling for Efficient Hand Activity Recognition .	88
4.1 Problem Statement and Motivation	88
4.2 Related Work	92
4.3 Study Design	99
4.4 THOR I: Adaptive Spatio-Temporal Sampling	102
4.5 THOR II: Hand-related Activity Recognition from a Patch	108
4.6 Results	113
4.7 Discussion and Future Work	120
4.8 Summary of Findings	126

Chapter 5: Enabling Gesture Customization with Minimal Training Data	128
5.1 Problem Statement and Motivation	128
5.2 Related Work	131
5.3 Method	134
5.4 Evaluation	142
5.5 Example Applications	150
5.6 Discussion and Future Work	154
5.7 Summary of Findings	157
Chapter 6: Conclusion	159
References	187

LIST OF FIGURES

2.1	<i>WildCam</i> system comprises an IR sensor array and an RGB camera. <i>WildCam</i> enables monitoring of eating behaviors and the environmental context related to eating, such as social presence.	20
2.2	Different activities captured by the camera. Blue boxes indicate eating, and red boxes represent social presence.	27
2.3	Activities (number of frames) captured by <i>WildCam</i> and their overlap. Green indicates the number of frames labeled as eating, and red indicates the number of frames labeled as containing social presence.	28
2.4	Functional overview of <i>WildCam</i> . Raw images are passed into the activity recognition models and used to manually annotate the ground truth. DBSCAN-based clustering is independently performed on the output of the activity recognition module and the manually annotated ground truth. A comparison of the clustering of the machine-learned (ML) model output is compared against the labeled ground truth's clustering output to determine the performance of each model.	30
2.5	Data augmentation for face: Using image processing techniques, normal images in a public dataset (a) were degraded. As a result, more training data (b) similar to the data from our dataset (c) were generated.	32
2.6	Sample cases causing false positives and/or false negatives in the system. a–d: Hand-to-mouth gestures that are confused with eating. e–g: Sensor position can affect the field of view captured by the IR sensor. Red boxes indicate field of view captured by the IR sensor.	42

3.1	SmokeMon is a chest-worn thermal-sensing wearable system with the lens pointed toward the wearer's mouth [A], capturing spatial, temporal, and thermal information around the wearer and cigarette [B] all day to unobtrusively and passively detect smoking events and estimates of smoking topography [C].	47
3.2	Thermal captures fine-grained information about the smoking session including: [A] cigarette ignition and [B] cigarette tip during a smoking session. Thermal, spatial, and temporal information can help in distinguishing between a non-puff (when the cigarette is in the FoV but not in the mouth [C]) and a puff (when the cigarette is near the mouth [D] or when there is no cigarette [E]).	60
3.3	Correlation (Pearson's $r=0.77$; $p<.01$) between the normalized ground truth puff volume obtained from CReSS Pocket and the one calculated from the thermal data.	60
3.4	Thermal sensors capture rich information that can help distinguish between many hand-to-mouth gestures.	61
3.5	Enclosure and attachment mechanisms in SmokeMon to maximize comfort while enabling reliable data collection.	62
3.6	SmokeMon framework to extract smoking topography. [A] We first extract possible smoking sessions based on thermal information. Then we [B] run a puff-detection model on each frame in the extracted session. Finally, [C] we perform postprocessing to extract puffing events and the smoking topography of the session.	65
3.7	In-lab experimental setup.	69
3.8	In-lab Bland–Altman plots comparing ground truth and SmokeMon smoking topography measurements.	75
3.9	In-wild Bland–Altman plots comparing ground truth and SmokeMon smoking topography measurements.	78
3.10	Overall, participants were not at all concerned with sharing their data.	83

- 4.1 Overview of the THOR method, a spatio-temporal RGB frame sampling method that uses thermal sensing to improve the efficiency of wearable camera systems. As illustrated in (A), the method computes a similarity score as a proxy for user activity engagement. Based on this score, it increases the temporal sampling rate during activity transitions (blue bounding boxes) and decreases it during periods of continuous engagement (green bounding boxes). The temporal sampling strategy (B) selects only the most informative frames, while the spatial sampling method (C) crops the image to focus on hand-object regions, reducing the overall volume of data to be processed. 89
- 4.2 Example activity categories and associated objects from our study (15 classes shown). Our method uses a thermal camera to monitor body position during these activities and samples a cropped RGB image around the hand and object-in-hand for hand-related activity recognition. In total, we collected over 30 activity classes in an in-the-wild study. 92
- 4.3 Overview of the THOR system. A wearable RGB camera is augmented with a downward-facing thermal sensor to capture hand and body positions. In THOR I (A), spatio-temporal adaptive sampling is achieved by processing thermal frames (A1) through a contrastive encoder to extract embeddings (A2), which are then compared via pairwise cosine similarity (A3) to compute a similarity score (A4) that dynamically adjusts the RGB frame rate. When RGB frames are captured, a patch centered on the hand and object-in-hand (A5) is extracted and passed to THOR II (B), which performs hand-related activity recognition. THOR II takes the patch (B1) and uses synthetic data to fine-tune a vision-language model (B2), generating free-form captions describing the user’s activity (B3). We evaluate THOR in terms of accuracy, latency, and battery lifetime. 99
- 4.4 In the 2D embedding space of consecutive thermal frames produced by THOR’s temporal sampling method, hand-related activity frames (blue circles) form clusters that are separated from transition frames (orange circles). The regions A–D mark the moments where new activities begin. The smaller isolated clusters (E) represent subtle intra-activity movements (for example, minor posture shifts while assembling a puzzle). 102
- 4.5 By using a low-resolution patch as input, the model processes fewer visual tokens compared to full-image input (A–B). In addition, fine-tuning the model to generate concise textual descriptions results in fewer language tokens (C–D), leading to reduced latency and lower power consumption on the device. 109

4.6	Coverage of segments with different length using different variations of THOR and baselines methods. All reported values are normalized with respect to their corresponding baselines. For instance, in (A), the percentage of segments captured by THOR is shown with Uni-High set as the 100% reference point.	113
4.7	Power consumption profile of different components in THOR I (A) and THOR II (B) compared to baseline methods.	115
4.8	Input (A) and end-to-end (B) Latency of THOR II compared to baseline method. .	115
4.9	Cosine similarity between narrations provided by annotators (A1 vs A2) and our method.	118
4.10	Accuracy of different THOR variants in classifying 30 hand-related activities, compared to using the full video. Note that each variant is designed for segments of different lengths; therefore, their accuracies should not be directly compared to one another. For example, the accuracy of THOR-low is calculated using only longer segments.	119
5.1	Overview of our method’s pipeline: We use a hand pose estimation model (A) to extract hand skeleton and landmarks. We utilize a graph transformer to capture both (B) joint-to-joint and (C) joint-to-group attentions. Temporal features are extracted using (D) temporal convolutions. We implement and enhance customization and few-shot learning (FSL) through (E) meta-augmentation and (F) meta-learning approaches.	131
5.2	Graph transformer architecture used in our method. The graph transformer takes the keypoints and joint groups (fingers), projects them into higher dimensions and calculates attention maps to extract spatial features. The spatial and temporal features are then aggregated through temporal convolutions to yield the final output.	135
5.3	Visualization of the transformer’s joint-to-joint attention maps for eight gestures. Each map is either a 21×21 (one-handed) or 42×42 (two-handed) matrix averaged across all attention heads ($N = 9$). The model attends to different parts of the hand based on the gesture. For example, for a <i>swipe right</i> gesture, most keypoints attend to the thumb and ring finger.	138

- 5.4 Shown here are the 20 gestures on which we evaluated our model. The blue square indicates the dynamic one-handed gestures, while the yellow diamond indicates the static one-handed gestures. Our dataset also comprises of six two-handed gestures, indicated by the green triangle for dynamic gestures and the red circle for static gestures. Our method can accommodate most gestures that a user might present from these gesture categories. This particular evaluation set was selected to represent the breadth of possible gestures, but should not be taken to suggest that our system can only accommodate these gestures. 140
- 5.5 Accuracy of our method and baselines when tested with (A) two, (B) three, and (C) four new gestures along with the background class. In (D) we show a specific experiment where our method achieves its maximum accuracy improvement over baseline models such as the random forest. In (E) we show a specific experiment where our method achieves its minimum accuracy improvement over baseline models such as the random forest. 144
- 5.6 The overview of our gesture design tool. (A) The designer starts with registering their own custom gestures by demonstrating each gesture once. (B) The designer can review the samples collected, visualize the landmarks, and decide to delete, keep or add samples. (C) A graph transformer model is trained on-device using the designer's collected samples in under a minute. (D) The designer can preview and assess the real-time performance of the model. Based on the result, the designer can decide to edit the gestures or samples by going back to step AB. (E) The custom gesture model can be further be exported to other devices or applications enabling seamless interaction and deployment. 149
- 5.7 (Left) An example video editing application where the user has customized a left-hand gesture to invoke the blade tool. (Right) The system also recognizes null classes and changes the tool back to the default pointer. 152
- 5.8 A virtual whiteboard where users draw with a pen-shaped hand (Left) and erase with a wiping gesture (Right). 153

LIST OF TABLES

2.1	Study summary. Percentages show each participant's share from total amount of frames labeled as that task.	29
2.2	Episode-level prediction result for eating episodes. RGB indicates that only RGB images were used for evaluation; RGB+IR indicates that both RGB and IR images were used for evaluation. # Episodes indicates the number of eating episodes that were present in the ground truth.	36
2.3	Episode-level evaluation result for social presence. RGB indicates that only RGB images were used for evaluation; RGB+IR indicates that both RGB and IR images were used for evaluation. # Episodes indicates the number of social episodes that were present in the ground truth.	38
2.4	Analysis of false positives and false negatives for activities and events monitored by WildCam as observed in our dataset.	40
3.1	Smoking topography measurements and their definitions	50
3.2	Comparison between thermal cameras available off the shelf	59
3.3	Current draw and power consumption of SmokeMon firmware under different settings: (1) sensing-only state: collects and processes the thermal data; (2) storage state: collects, processes, and stores the data on an SD Card; and (3) BLE state: collects, processes, and streams the thermal data over BLE.	64
3.4	Puffing statistics obtained from CReSS Pocket	71
3.5	In-wild participants self-reported demographics and number of cigarettes smoked per day and in-wild participants smoking data and topography summary	74

3.6	In-lab puff detection and puff count per participant	76
3.7	In-wild average puff prediction per participant, puff count, and smoking session detection. P10 smoking session detection analysis was removed because the ratio of smoking events to nonsmoking events was not similar to other participants because the participant did not follow the study protocol and turned the device on during smoking sessions only.	78
3.8	Further energy optimizations that can be achieved with SmokeMon. P10 is not included in this analysis because ratio of smoking events to nonsmoking events is not similar to other participants due to the participants not following the study protocol.	80
4.1	Percentage of data utilized per participant for each THOR configuration.	112
4.2	Power consumption of different components of the wearable device.	116
4.3	Inference energy consumption of iPhone 15 Pro under different settings.	116
4.4	Device battery lifetime using different methods.	116
4.5	Precision (P), Recall (R), F1-score (F1), and Accuracy (Acc) for THOR II and baseline methods.	118

CHAPTER 1

INTRODUCTION

The human hand is a remarkable instrument of action and expression, central to how we interact with the world and each other. From the fundamental acts of survival, such as eating, to complex social behaviors and health-related habits like smoking, our hands are constantly engaged in a rich tapestry of activities. The ability to automatically recognize and interpret these hand-related activities holds immense potential for advancing both personal health monitoring and seamless user experiences. Behaviors such as overeating and smoking remain major contributors to preventable disease and mortality. Understanding and monitoring these behaviors in daily life is therefore an important step toward designing interventions that can improve long-term health outcomes.

Wearable computing has advanced rapidly in the past decade, enabling rich and continuous sensing of human activity [1]. Motion sensors, for example, play a crucial role in providing strong signal for physical activities [2]; however, their data are abstract, often ambiguous, and difficult to interpret. For example, hand-to-mouth actions such as eating and smoking appear similar when relying solely on inertial data. Cameras, on the other hand, provide detailed contextual cues that reveal how the hand interacts with objects and the surrounding environment [3]. This information can support applications such as detecting eating, smoking, or medication intake, as well as recognizing gestures in immersive computing systems [3]. Cameras also serve as a valuable source of ground truth in multimodal sensing studies, which is why synchronized video is frequently collected for annotation and labeling purposes [4].

However, the vision-based approach comes with formidable challenges that limit its widespread deployment in everyday life. Continuous high-resolution RGB video capture drains device batter-

ies rapidly, generates excessive data volumes, and poses major privacy concerns for both users and bystanders. Furthermore, real-time video processing demands computational resources that are often infeasible for lightweight, embedded platforms. These limitations reveal a fundamental tradeoff in wearable vision systems, where increased visual detail and informativeness come at the expense of long-term sustainability and privacy.

This dissertation addresses these challenges by adopting the Principle of Least Privilege (POLP) for sensing. The core idea is to collect only the minimal amount of information necessary for accurate recognition. The central hypothesis is that not all sensing modalities or moments in time are equally informative for understanding hand-related activities. By selectively capturing and processing only the most valuable information, it is possible to achieve efficiency, privacy, and accuracy simultaneously. To operationalize this principle, the dissertation develops systems and algorithms that optimize how vision-based sensing is performed and interpreted in wearable scenarios. The work centers on using a low-power thermal sensor as the primary source of information, reducing reliance on the more power-intensive RGB sensor while preserving utility for hand-activity recognition. First, the work investigates how a low-power thermal sensor can be utilized to detect hand-to-mouth gestures such as eating and smoking (Chapter 2 and 3). Second, it examines how thermal cues can guide adaptive RGB sampling to maintain recognition performance while reducing power and computational costs in wearable hand-activity recognition (chapter 4).

The remainder of this document is structured as follows. Chapter 2 presents WildCam, a dual-stream wearable camera that integrates low-resolution thermal sensing with RGB imagery to study eating and social presence in real-world settings. The chapter demonstrates how thermal cues enhance visual recognition while improving privacy and energy efficiency, and it discusses challenges posed by motion patterns, occlusions, and environmental variation. Chapter 3 introduces SmokeMon, an energy-efficient thermal-based system for detecting smoking behavior and quantifying

smoking topography, including puff count, duration, and intensity. Unlike RGB-based approaches, SmokeMon relies exclusively on thermal imagery, enabling unobtrusive and privacy-preserving monitoring in both laboratory and free-living environments. Chapter 4 presents THOR, a thermal-guided multimodal framework for hand-object recognition. THOR dynamically adjusts RGB sampling based on thermal signals, capturing high-frame rate RGB data only during salient events such as object engagement or gesture initiation. This strategy significantly reduces power consumption and data volume while maintaining recognition accuracy. The chapter also introduces a synthetic-augmented teacher–student method for robust hand-object recognition under partial occlusion or limited visibility. Chapter 5 addresses the challenge of reducing the cost of model training and personalization. It introduces a vision-based gesture customization framework that enables users to define personalized gestures from a single visual demonstration. Through the use of a graph transformer and meta-learning, the system adapts to both static and dynamic gestures across egocentric and allocentric views, lowering data requirements and enabling rapid on-device training. Chapter 6 concludes with a synthesis of insights, discusses implications for future wearable systems, and outlines directions for extending adaptive sensing to broader domains.

CHAPTER 2

LOW-POWER THERMAL CUES FOR AUGMENTING RGB-BASED VISUAL SENSING

2.1 Problem Statement and Motivation

A recent study found that strategies for preventing dietary lapses may differ based on social presence (*i.e.*, the presence of another individual during the eating episode), which is known to influence many health habits [5]. For example, eating with a companion was shown to increase food intake by 28% [6]. On the other hand, one possible criterion for defining a binge eating episode (overeating while sensing loss of control) is eating alone because of feeling embarrassed by how much one is eating. Human behavior is complex and multifactorial, and it is necessary to understand the interplay between eating and social presence to model their relationship more accurately within conceptual models of human behavior.

Determinants of eating behaviors and the importance of the social context in which the behaviors occur have been a long-standing interest for researchers [7], [8], [9], [10], [11]. These contexts have been studied together in laboratory conditions [8], [9], [11], [12], [13]; however, there is an unmet need for studies in real-world settings as in-laboratory studies cannot be generalized to real-life owing to the myriad social and environmental factors that affect human behaviors [14], [15], [16], [17], [18], [19], [20].

With the advancement of wearable computing, several devices have emerged that can automatically capture and model human behaviors, such as wrist and neck sensors. However, wearer adherence to multiple devices in the wild is a known problem, preventing the practical utility of these devices in longitudinal studies. Expecting participants to wear more than one device reliably

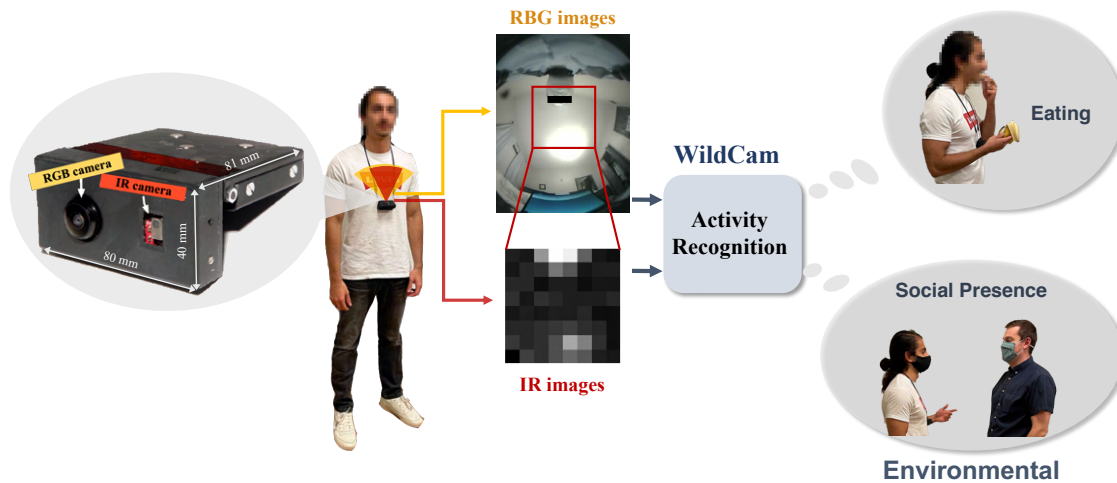


Figure 2.1: *WildCam* system comprises an IR sensor array and an RGB camera. WildCam enables monitoring of eating behaviors and the environmental context related to eating, such as social presence.

reduces the feasibility of longitudinal studies while simultaneously increasing chances for multiple points of device failure [21] and challenges with synchronization [22]. One promising device that can monitor multiple human behaviors is a wearable camera. Wearable cameras can provide visual confirmation of human activities, and advances in image processing and machine learning models can automatically detect whether an individual is performing an activity such as eating or is in the presence of others (Figure 2.1).

The camera's benefit over other sensing modalities is the ability to visually confirm and validate activities such as eating [23], [24] or screen time [25]. Researchers have used these cameras to visually confirm several behaviors, including an individual's physical activity [26], [27], [28], sleep pattern [29], [30], and more recently eating behaviors [31], [32], [33]. SenseCam, one of the earliest cameras used to confirm human behavior in the wild visually, captured one image every 30 seconds and built a memory log for the wearer but had a storage limitation of 1 GB [26]. This intermittent capture of behavior helps ensure an all-day battery lifetime; however, the inability to

continuously record video prevents its utility in capturing fine-grained behavior related to eating (*e.g.*, feeding gestures) and social presence (*e.g.*, someone coming into and out of the field of view). This fine-grained capture is needed not only for timely visual confirmation but also for automated detection of activity [34].

Moreover, while SenseCam was capable of egocentric video capture, its lack of mechanical customization made it challenging to capture certain wearer activity, particularly for varying body shapes and sizes. For some activities, like detecting feeding gestures for eating, it increases reliability when the camera is oriented towards the wearer’s mouth. Activity-oriented cameras are particularly well suited to capture a specific activity of the wearer as they are designed and oriented to capture the wearer [35]. Participants can conveniently wear activity-oriented cameras to collect fine-grained activity data (*e.g.*, hand-to-mouth gestures, eating, food types consumed) and environmental context associated with the specific activity. However, most wearable cameras tested in detecting feeding gestures are high-resolution cameras that consume significant amounts of energy and prevent the capture of all-day activity. The use of low-resolution RGB video is necessary to address power constraints without significantly increasing battery weight and size; however, low-resolution RGB video may compromise the machine learning model’s ability to reliably detect the behavior of interest [36]. Recent literature has shown the *potential for thermal imagers to augment RGB data, thereby enhancing a camera’s ability to detect salient objects* [37]. Simultaneously, to mitigate power constraints, researchers have shown the ability of low-resolution thermal imagers to detect human activities such as lying, sitting, standing, and walking [38].

A thermal sensor provides unique information about objects with thermal signatures (*e.g.*, hot and cold objects), including the human body. For instance, thermal cameras can successfully capture a person’s silhouette as the average body temperature is usually higher than the surrounding temperature. The use of thermal images for human activity recognition via wearables has become

increasingly common in recent years, as low-resolution, power-efficient thermal cameras are now available [39], [40], [41]. A further application of thermal images is to facilitate the segmentation of human silhouettes in RGB images to enable privacy-preserving applications such as face obfuscation [42], [43]. More recently, researchers used thermal sensors to reduce the power consumption of wearable devices by developing a triggering mechanism that uses thermal images to execute a high-power-consuming task only when needed [44].

In this chapter, we focus on the ability of a low-resolution RGB and thermal sensor to detect two critical activities of interest – eating gestures and social presence.

The **key contributions** of this work are:

1. **Activity detection using a low-resolution, wearable activity-oriented camera comprising RGB and thermal sensing:** We present a framework combining a low-resolution, wearable activity-oriented camera with a low-resolution IR sensor array *as a tool* and corresponding data analytic pipeline to detect eating gestures and social presence. Based on a study with 10 participants and 80 hours of video across 3 days of device wear, we detect eating episodes and social presence with an F1-score of 70% (5% increase using IR) and 74% (44% increase using IR), respectively. In addition, we report results with and without the IR sensor array.
2. **Implications for device design and real-world studies:** We study the false positives and false negatives for each activity and provide guidance on the design of future multimodal activity-oriented devices for detecting eating and social presence and for deploying such devices in real-world studies (see section 2.6).

2.2 Related Work

This section discusses recently developed methods and systems to automate the detection of eating episodes and faces/social presence. We primarily focus on image-based approaches. Additionally, since we use an IR sensor array, we provide a brief background of the promise of this sensor in capturing our behaviors of interest.

2.2.1 Automated Eating-Detection Approaches

The ability to automatically detect eating in free-living settings has been a long-standing research problem in the mobile and wearable computing community. Over the years, researchers have explored various proxies to detect eating episodes, such as the use of inertial sensor data from a smartwatch to detect feeding gestures [45], [46], [47], [48], audio sensor data to detect chewing or swallowing sounds [31], [49], [50], cameras to capture images of food items [51], [52], [53], and a variety of other sensing techniques [33], [54], [55]. Although inertial-based sensors can detect feeding gestures, automated models are often confounded by similar signals from other hand-to-mouth gestures [56], and they are unable to provide data regarding the presence of other individuals. Similar to inertial-based sensors, audio-based methods have shown promise in detecting eating activity, but they provide little utility in detecting social presence.

More recently, with the availability of wearable cameras, visual confirmation of behavior became possible to monitor fine-grained details of the eating activity. This is advantageous for the rich sensor information in video signals about an eating episode’s characteristics, allowing for the detection and confirmation of multiple behaviors. However, annotating and coding the eating activity based on camera visuals is a timely, costly, and labor-intensive process. Methods that automatically detect these details will significantly improve our ability to design studies that understand factors

contributing to overeating in real-world environments.

High-resolution image- and video-based methods have been explored by several researchers to detect feeding- and eating-related gestures. Reddy et al. [57] and Thomaz et al. [53] suspended a smartphone across the neck using a lanyard to capture images throughout the day, including moments of eating. However, suspending a smartphone across the neck can be uncomfortable for long-term wear. To overcome this, Bedri et al. [51] designed FitByte, a smartglass with an embedded camera, and Echterhoff et al. [58] designed the personal activity radius (PAR) device that could be attached to any glass frame. In addition, both devices used a camera to monitor and detect eating moments. More recently, Jia et al. [59] used the eButton device to capture egocentric images of food being consumed. To enable data collection from individuals in a privacy-preserving approach, Schiboni et al. [60] designed a head-mounted camera system that points downward from a cap and captures eating moments without affecting user privacy. Recently, Rouast and Adam [61] used a 360° video camera placed at the center of a dinner table to record participant meals and built feeding gesture-detection models using deep neural networks. However, this approach is limited to a controlled laboratory setting. In contrast, our approach adopts a low-resolution and low-power, wearable activity-oriented device [35] with a fish-eye lens (oriented toward the mouth), which enables visualization of food and drink from table to mouth.

2.2.2 Automated Social-Detection Approaches

To detect social presence, one approach is that of face detection, a specific type of object-class detection that has been an important part of image processing and machine learning research in the past three decades [62]. Shetty et al. [63] have shown the utility of detecting faces using a thermal sensing array, the “Grid-EYE,” combined with a Kalman filter to detect human motion. Other modes of detection have also shown promise, including RFID tags [64], [65], Bluetooth Low En-

ergy [66], wearable sensor (GNSS/IMU)-based human tracking and motion capture [67], and even WiFi signals, which enable detection of people from behind walls [68]. These modalities are useful but are either too costly or not wearable and hence are limited to a specific region, require access to devices on multiple individuals, or are unable to provide visual confirmation of the behavior, thereby preventing validation of the activity of interest.

Although sensing faces has been a long-standing research topic, given the ubiquitous nature of video cameras, understanding social presence and interaction are of more recent interest. In one of the earliest *egocentric* camera-based social interaction studies, Fathi et al. [69] used a Go-Pro camera to detect and characterize social interactions by focusing on assigning roles to individuals engaged in a conversation and localizing them in a 3D scene. Alternatively, Alletto et al. [62] clustered people into different socially related groups using a novel head pose estimation framework and then localized them in a 3D scene. In both of these studies, the researchers assumed that people would likely look at each other, whereas in the real world people can interact without constantly orienting themselves toward each other. These studies primarily performed classification on pre-segmented interaction clips, rather than identifying interactions holistically by detecting and segmenting them in free-living settings. In addition, Zhao et al. [70] used high-resolution images from multiple chest-worn cameras to analyze human interaction. However, this study did not focus on capturing the wearer’s social activities. All of these studies used a Go-Pro or similar high-resolution camera. Go-Pro (and similar) cameras provide high-definition images with several pixels on target where substantial detail can be captured, simplifying the task of face detection. A recent study by Okuno et al. [71] counted the number of faces to build an index that quantifies the amount of social activity; however, they still relied on high-quality RGB images and used out-of-the-box face detection models to take the initial step. Continuously capturing images at a high resolution is battery intensive—the Go-Pro lasts only a few hours [72]. Since we are interested in

capturing interactions throughout the day, we investigated using lower-resolution images [73].

2.3 System Design

2.3.1 Activity Definitions

We defined two constructs, *eating episode* and *social presence*, and we describe each construct and the method used to annotate each of these constructs below. Two trained annotators independently annotated each video frame and then met to resolve any conflicts. Remaining conflicts were resolved in a group meeting with expert annotators. Figure 2.2 presents sample images captured by the WildCam system.

Eating episode: We defined feeding as an activity involving a hand-to-mouth gesture where the hand brings food to the mouth. The gesture starts when the hand (with food) begins traveling toward the mouth (seen in the images) and ends when either (1) the hand returns to a position of rest, (2) the hand is no longer visible in the captured images after putting the food in the mouth, or (3) the participant immediately starts the next feeding gesture by picking up the next morsel of food. Eating was determined by the annotator viewing the image (food or hand visible in the low-resolution image and act of eating present).

An eating episode was a combination of one or more feeding gestures, each separated by no more than 15 minutes (as defined by recent studies [31], [33]) from adjacent feeding gestures.

Social presence: We defined the wearer to be "alone" when no other visible face was present in the scene, as captured by the wearable camera. Social presence was defined as when one or more faces other than the participant's were visible in the scene captured by the wearable camera. We considered a person's face to be visible if at least one eye could be visually confirmed by the annotator. For cases where the back of the head of a bystander was seen, we assumed the bystander

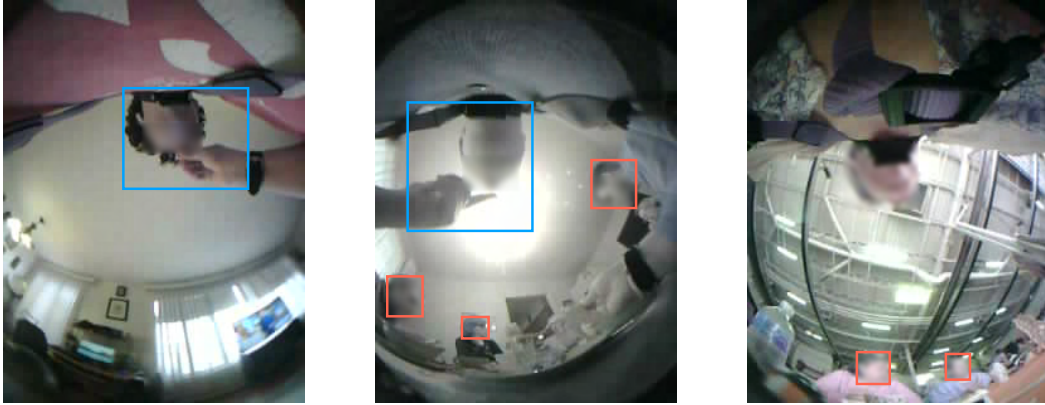


Figure 2.2: Different activities captured by the camera. Blue boxes indicate eating, and red boxes represent social presence.

was not engaged with the wearer, and we did not consider that face to be visible, thus lacking social presence. This could become one of the limitations of our approach, which we address in Section 5.6.

2.3.2 Hardware Apparatus

We developed WildCam (Figure 2.1), an activity-oriented camera, which is the modified version of an activity-oriented camera system that has been previously envisioned by Alharbi et al. [42]. WildCam is a dual-stream sensing video camera with a fish-eye lens that records the wearer’s activities and part of the surrounding environment. WildCam is built around an STM32L4 microcontroller (ARM Cortex-M4), with two imagers: an RGB camera without IR filter (OmniVision OV2640¹ with an 180° fish-eye lens), and a 8×8 IR sensing array (GridEYE AMG8833). We also add night vision capability via an IR illumination LED. A user of WildCam wears the device in such a way as to ensure that their head is partially captured in the IR sensor’s field of view. The Panasonic GridEYE AMG8833 IR array has a narrower field of view (*i.e.*, 60°) focused in the upper portion

¹Arducam OV2640 https://www.arducam.com/downloads/modules/OV2640/1Inch4_2_Megapixel_OV2640_CMOS_Camera_Module_REVB_DS.pdf

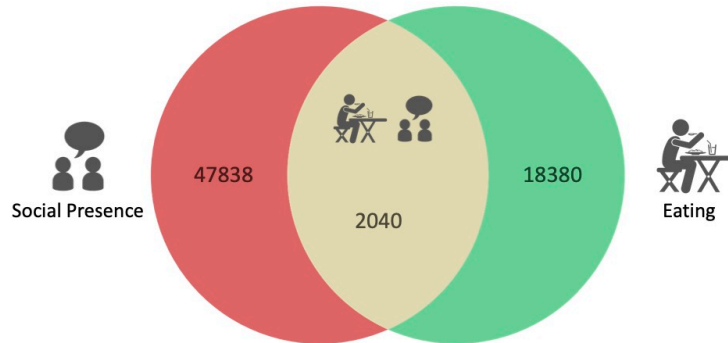


Figure 2.3: Activities (number of frames) captured by WildCam and their overlap. Green indicates the number of frames labeled as eating, and red indicates the number of frames labeled as containing social presence.

(facing the user) of the OV2640 view, allowing us to capture hand-to-head-related gestures (*i.e.*, eating). WildCam gathers the raw video and IR data streams at 5 frames per second, encrypts these streams on the fly using a stream cipher (salsa20 [74]), and stores these images onto a microSD card. We extracted and processed the data streams offline when the participants returned the device.

2.3.3 Participants and Study Dataset

We recruited 10 participants (5 male, 5 female; mean [SD] age, 45.2 [14.8] years). Participants wore WildCam for 3 days, during which we collected data from the device. This study was approved by the Northwestern Institutional Review Board, and all participants provided written informed consent before enrolling in this study. As we were interested in understanding the eating behavior of individuals with obesity, we recruited participants with a body mass index >30 kgs/m². Although recent studies have collected data with larger sample sizes, there are limited data for individuals with obesity. Recent work has shown that models developed using data from adults without obesity perform poorly for adults with obesity [33], thus making a case for building models with data from

Table 2.1: Study summary. Percentages show each participant’s share from total amount of frames labeled as that task.

	Participant	Eating	Socializing
First Fold	P1	14%	50%
	P2	7%	0%
	P3	2.3%	0%
Second Fold	P4	9.8%	8.5%
	P5	19.7%	5.6%
	P6	5.1%	3.1%
	P7	11.4%	7.2%
Third Fold	P8	7.3%	11.4%
	P9	8.2%	11%
	P10	14.7%	2.8%

participants with obesity.

Participants were instructed to wear the WildCam device throughout the day while doing their everyday activities. We instructed participants to ensure the IR sensor could capture their faces while wearing the device by pointing the lens upward toward their chin. To ensure uniformity of data across participants, we randomly selected 8 unique hours of data from each participant across the 3 days, which resulted in labeling and analyzing 80 hours of data in the wild. Overall, our dataset had *1,440,000 frames* of RGB and IR sensor images, and each frame in the video was labeled to indicate whether the participant was (1) eating or not eating and (2) alone or not alone in that frame. Figure 2.3 shows the distribution of the images across the two activities.

2.4 System Processing Framework

2.4.1 Overview

Figure 2.4 presents a functional overview of WildCam. Our proposed framework comprises the following stages:

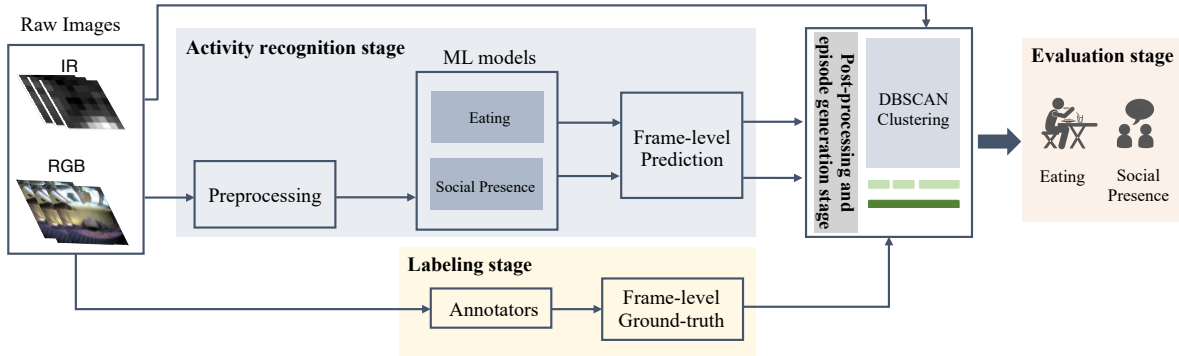


Figure 2.4: Functional overview of WildCam. Raw images are passed into the activity recognition models and used to manually annotate the ground truth. DBSCAN-based clustering is independently performed on the output of the activity recognition module and the manually annotated ground truth. A comparison of the clustering of the machine-learned (ML) model output is compared against the labeled ground truth’s clustering output to determine the performance of each model.

- 1) **Labeling:** Annotators label whether there is a face and/or eating gesture at the frame level (discussed in Section 2.3.1).
- 2) **Activity recognition:** Our system augments the RGB images with the IR sensor data and trains a model to predict (either a face or eating gesture) at the frame level using RGB data. Section 2.4.2 provides details of the activity recognition stage.
- 3) **Post processing and episode generation:** Our system uses the IR sensor array to remove false positives for each activity. At this stage, WildCam applies clustering techniques to generate start and end times of both ground truth and predicted events. Details of the episode generation stage and IR utility are provided in Section 2.4.2.3.
- 4) **Evaluation:** We evaluate the performance of our system by comparing the results of the machine learning approach and the ground truth at the episode level. Details of the evaluation are provided in Section 2.5.

2.4.2 Activity Recognition

We performed the activity recognition task independently for each of the two activities. For each activity, the overall activity recognition task comprised a preprocessing stage, a machine learning model creation stage, and a postprocessing and classification stage to generate episode level predictions.

2.4.2.1 Preprocessing

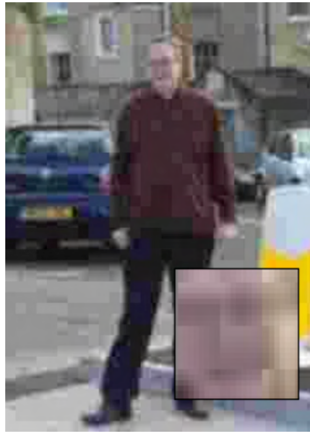
The first step in detecting activities and context such as eating episode or social presence is to preprocess the raw images. The following steps were taken on the images for each of the activities and context recognition.

Image size and rotation (used in social presence recognition): The WildCam hardware captures RGB images of 320×240 pixels. We tested each image at two scales, the original and double scale (i.e., 640×480 pixels). We realized that resizing the image had a substantial positive effect during inference, mainly due to the enlarged faces. Moreover, due to the distortion properties of a fish-eye lens, objects captured by the lens may not be oriented properly in the image (Figure 2.2). Therefore, this adds a layer of difficulty in detecting faces and objects at different orientations, which most face detection models fail to address. To address this, we rotated each image by 45° , 90° , 135° , and 180° and used the original image and the four rotated versions of it in creating the face detection model that is described later.

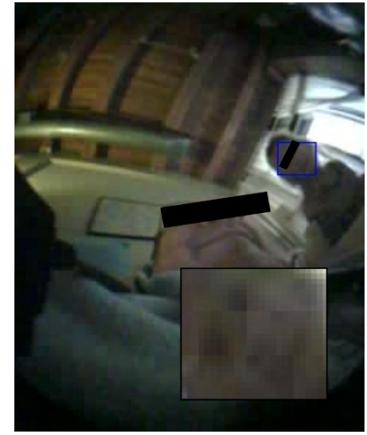
Dataset augmentation (used in social presence recognition): Training data for the social presence recognition model consisted of data from 10 participants and around 37,000 images labeled with a bounding box around the faces. Although this sample size appeared to be enough to train a face detection model, there was low diversity in identity and facial characteristics in the dataset. To this



(a) Sample face from WIDERFACE



(b) Processed face from WIDERFACE



(c) Sample face from our dataset

Figure 2.5: Data augmentation for face: Using image processing techniques, normal images in a public dataset (a) were degraded. As a result, more training data (b) similar to the data from our dataset (c) were generated.

end, the training set was augmented with WiderFace [75], a large face detection dataset with a high degree of variability in scale, pose, and occlusion. Images included in this dataset are mostly high-resolution images with more details compared with WildCam images. To account for this, we first modified images in the WiderFace dataset by using an averaging filter of size of 4×4 to simulate blurriness caused by the motion. Next, we performed the same JPEG compression used on-device with WildCam to diminish more details from faces and create similar faces in our dataset (Figure 2.5a, 2.5b and 2.5c).

Optical flow calculation (used in eating activity recognition): Eating gesture is not an instantaneous activity but includes temporal information that one has to consider for reliable detection performance. One approach to quantifying this temporal information is by using optical flow. We relied on the deep-learning optical flow estimator FlowNet2², which can process up to 140 frames per second when estimating the optical flow.

²<https://github.com/NVIDIA/flownet2-pytorch>

2.4.2.2 Machine Learning Model Creation

The preprocessed data are then fed into a machine learning model that can recognize specific activities.

Eating activity: As mentioned before, there are two levels at which we detected the eating activity: (1) at a per-frame level and (2) at an episode level (described in Section 2.4.2.3). For the per-frame level, our model aimed to determine whether a participant was performing an eating gesture in frame F_i . We used a two-stream CNN architecture to detect the eating gesture in each frame, where the first stream represents the spatial channel and the second stream the temporal channel [76]. The spatial and temporal channels were trained independently with ResNet101 as the backbone.³

To train the network, we used a binary cross-entropy loss function. To address the class imbalance problem, we leveraged the approach used by Rouast et al. [61]. That is, in each mini-batch, we scaled the loss for class i by w_i according to:

$$w_i = \frac{m}{C(i) * n}$$

where m is the number of labels ($y = \{y_1, \dots, y_m\}$), n is the number of classes, and $C(i)$ is the number of elements of y that equal y_i .

Overall, the model classified the i^{th} frame as eating or not eating based on a context of 12 other frames surrounding the target frame. The spatial stream accepted a single $320 \times 240 \times 3$ frame F_i as input and classified it as eating or not eating. The temporal stream accepted stacked optical flows from the sequence of 13 input frames $[F_{i-6}, F_{i+6}]$ to classify the F_i frame as eating or not eating. The final classification was found by late fusion in averaging the outputs of the spatial and

³<https://github.com/bryanyzhu/two-stream-pytorch>

temporal channels as suggested by Simonyan et al. [76].

Social presence: Due to the camera’s low-resolution output and distance to bystanders, WildCam mainly captures tiny faces. Moreover, motion artifacts and the camera’s on-device JPEG compression further reduce important details from the images, posing a challenge to the face detection task. We used RetinaFace [77], a tiny face-detection model that employs extra supervision and multitask learning to handle faces across a wide range of scales. RetinaFace achieves a mean average precision of 91.4% on WiderFace, demonstrating strong performance on tiny-face benchmarks.. However, this model was trained on the WiderFace [75] dataset, which comprises higher-resolution images and differs significantly from our dataset. Figure 2.5a and 2.5c shows sample images from WiderFace and our dataset. To account for the differences, we retrained RetinaFace with degraded versions of images from the WiderFace dataset and additional labeled RGB images from our camera’s RGB images. More details regarding dataset augmentation and preprocessing methods can be found in Section 2.4.2.1. All hyperparameters in the architecture and weights of the backbone network (*i.e.*, Resnet-152), were kept the same as in the original work to keep the training process short. Using our model, we classified an image as having a face (*i.e.*, not being alone) when there was at least one detected face above a confidence threshold across the five possible combinations of orientations; otherwise, the image was classified as alone.

2.4.2.3 *Episode Generation: Postprocessing and Classification*

The final step in our system pipeline is the step that enables clustering data from the activity recognition stage to create activity-related episodes. However, before creating the episodes, the data are postprocessed and cleaned as described below.

Postprocessing: The IR sensor is capable of detecting the heat emitted by humans and objects in close vicinity to the camera. We utilized this capability of the IR sensor array to remove false posi-

tives (*i.e.*, falsely detected humans and human-related activities) generated in the machine learning step described in Section 2.4.2.2, thus increasing the precision of the detected activity or event.

Episode generation: Several researchers have generated episodes of activities and events to understand the specific temporal order of such activities/events. For eating activities, a number of researchers focused on evaluating eating episodes [33], [54], [78]. We focused on creating similar episodes by using a density-based approach (DBSCAN) on frame-level ground truths and predictions. This method required two parameters, namely eps and min_{pts} . Parameter eps was defined as the minimum distance needed between two clusters for the clusters to be merged into one, and min_{pts} was defined as the minimum number of points required for a cluster to be created.

Eating episode and postprocessing: We used a density-based (DBSCAN) approach, similar to one proposed by Zhang et al. [79], to create eating episodes. Similar to prior works, we used a 15-minute gap between gestures as the boundary between eating episodes [31], [79]. At the prediction stage, frame-level outputs were clustered with the same parameters used for ground truth to prepare predicted episodes for evaluation. We then used the IR data from each predicted episode to decide if it was a false positive. An eating episode consists of multiple hand-to-mouth gestures that can be captured and identified by the IR sensor array. Using this fact, false positive eating episodes initially detected by the RGB images were filtered consequently because of the absence of IR activity.

Social presence episode and postprocessing: Episode-level predictions were achieved by clustering frame-level predictions via DBSCAN. Since there is no clear definition of an episode with or without social presence, we took a data-driven approach to determine DBSCAN parameters. First, we calculated the cumulative distribution function (CDF) of the number of continuous frames with faces (a proxy for min_{pts}) and the number of frames between each set of successive frames with faces (a proxy for eps). We then identified the saturation point using a technique introduced by Ville

et al. [80] for each CDF and chose min_{pts} and eps accordingly. Due to the camera’s orientation and fish-eye lens, the wearer’s face was visible in the frames and was confounded with bystanders’ faces. As a result, we benefited from the IR sensor as a postprocessing step to estimate the position of the wearer’s head and filter false positive faces that overlapped with the wearer’s head. Our IR sensor captured the participant’s head in its small field of view; using the image, a mask was created on the participant’s head location by thresholding the body’s temperature. This was done by Otsu’s method [81], which automatically bins pixels into two classes based on pixel intensity. In our case, the classes were human and background. Then, overlap was calculated between the human mask and the bounding box area around the predicted face. The face was discarded if the overlap was above a threshold (empirically set to 200 pixels).

2.5 Evaluation

	RGB			RGB+IR			# Episodes
	Recall	Precision	F1-score	Recall	Precision	F1-score	
P1	0.5	0.4	0.44	0.5	0.5	0.5	5
P2	0.75	1	0.86	0.75	1	0.86	9
P3	0.67	0.5	0.57	0.67	1	0.8	3
P4	0.67	0.75	0.71	0.67	1	0.8	10
P5	0.86	0.75	0.8	0.86	0.86	0.86	8
P6	0.71	0.63	0.67	0.71	0.83	0.77	7
P7	0.56	0.63	0.59	0.56	0.63	0.59	9
P8	0.67	0.44	0.53	0.67	0.5	0.57	6
P9	0.8	0.5	0.62	0.8	0.57	0.67	5
P10	0.56	0.71	0.63	0.56	0.71	0.63	10
Total	0.67	0.63	0.65	0.67	0.74	0.7	72

Table 2.2: Episode-level prediction result for eating episodes. RGB indicates that only RGB images were used for evaluation; RGB+IR indicates that both RGB and IR images were used for evaluation. # Episodes indicates the number of eating episodes that were present in the ground truth.

2.5.1 Dataset Split

When splitting the data for training and testing purposes, we considered the variability in how people spend their time on different activities in a real-world daily-living situation. Figure 2.3 shows the imbalance in distribution of activities in our dataset, with some participants (*e.g.*, P2, P3) having almost no social presence and some (*e.g.*, P1) having approximately 50% of their total time being social presence. Due to this imbalance, common dataset split approaches such as person-independent training, validation, and testing might not have been appropriate. To this end, we split the dataset into three folds by manually picking participants such that there was sufficient data for each activity in each fold. Table 2.1 shows participant distribution in each fold to create three balanced splits. We ensured that we did not bias our testing process by performing cross-validation.

2.5.2 Evaluation Metric

For each activity and event, we were interested in understanding the prediction capabilities of the RGB images and observing improvement in results with the introduction of the IR sensor array. Thus, our evaluation metric revolved around the performance of detecting activities using RGB-only data and RGB+IR data at an episode level. Specifically, for eating episodes and social presence, we computed the precision, the recall, and the F1-score of detecting the episodes. This computation was done separately for RGB only and RGB+IR.

2.5.3 Episode-Level Evaluation

We compared our prediction approach against the ground truth to determine WildCam’s performance. False positivess occur when WildCam incorrectly predicts an episode when there is no episode, whereas false negatives occur when an episode present in the ground truth is not predicted.

	RGB			RGB+IR			# Episodes
	Recall	Precision	F1-score	Recall	Precision	F1-score	
P1	0.67	0.3	0.41	0.67	0.86	0.75	9
P2	-	-	-	-	-	-	0
P3	-	-	-	-	-	-	0
P4	1	0.17	0.29	1	0.75	0.86	3
P5	0.5	0.17	0.25	0.5	0.5	0.5	2
P6	1	0.06	0.11	1	1	1	1
P7	1	0.4	0.57	1	1	1	2
P8	0.75	0.33	0.46	0.5	0.8	0.62	8
P9	1	0.33	0.5	1	0.89	0.94	7
P10	1	0.07	0.13	1	0.29	0.44	1
Total	0.82	0.18	0.3	0.77	0.71	0.74	33

Table 2.3: Episode-level evaluation result for social presence. RGB indicates that only RGB images were used for evaluation; RGB+IR indicates that both RGB and IR images were used for evaluation. # Episodes indicates the number of social episodes that were present in the ground truth.

By comparing the occurrence of events in the ground truth episode and the prediction, we determine if a false positive or a false negative has occurred. If an episode is present and overlaps the ground truth and prediction, then it is a true positive. A true positive can span multiple ground truth events, or multiple true positives can occur for a single ground truth entry. We sum all true positives, false positives, and false negatives and use these values to compute precision, recall, and F1-score.

Eating detection: Overall, there were 72 eating episodes in our dataset (ground truth). When we used the RGB-only approach and performed three-fold cross-validation, we attained a recall, precision, and F1-score of 67.16%, 63.38%, and 65.21%, respectively. Table 2.2 shows the performance for each participant independently. For one participant, we could precisely detect all recalled eating episodes. We next computed the performance of WildCam when we included the IR sensor as a postprocessing step. Table 2.2 also presents the performance of detecting eating episodes when we used the RGB+IR approach. There was a substantial increase (>10%) in the precision, indicating

that the IR sensor successfully removed several false positives. Using the RGB+IR approach, the updated recall, precision, and F1-score were 67.16%, 73.77%, and 70.31%, respectively. Using the RGB+IR approach, for three participants, at a per-person level, we could achieve 100% precision, whereas for all participants, the precision and F1-score remained the same or improved compared with the RGB-only approach. Ultimately, the IR sensor helped remove 11 false positive episodes from the prediction.

Social presence: In our dataset, 33 episodes indicated social presence. However, for two participants, there were no social events at all. Overall, of these 33 episodes, we could detect 27 episodes using the RGB approach. Table 2.3 presents the per-participant performance. Overall, we attained a recall, precision, and F1-score of 81.81%, 18.12%, and 29.67%, respectively. We had excessive false positives, which adversely affected our precision and F1-score and introduced substantial noise in our prediction.

However, with the introduction of the IR sensor in conjunction with RGB, we substantially reduced these false positives. As a result, the precision improved from 18.12% to 71.05%, an approximately 53% improvement. Due to this improvement in precision, the overall F1-score also improved to 73.97%. Thus, the IR sensor successfully captured the wearer’s head and face and distinguished the wearer’s face from the bystander’s face.

Summary: Overall, the RGB-only approach of detecting eating episodes and social presence was able to attain reasonable performance. More importantly, eating episode and social presence detection can substantially benefit from using an IR sensor array in its postprocessing step. Indeed, the precision for detecting social presence increased by almost 53% when the IR sensor data were used, thus showing the benefit of the RGB+IR approach.

	False Positive Cases		False Negative Cases	
	Reason	Share	Reason	Share
Eating Episode	Confounding hand gestures	0.57	Drinking gestures	0.9
	Device motion artifacts	0.43	Partial occlusion	0.1
Social Presence	Video calls	0.19	Tiny faces	0.69
	I see ghosts	0.32		
	I see myself	0.31		
	Mirror mirror on the wall	0.19	Sensor design	0.31

Table 2.4: Analysis of false positives and false negatives for activities and events monitored by WildCam as observed in our dataset.

2.6 Implications for Design: An Assessment of False Positives and False Negatives

In this section we further elaborate on false positive and false negative cases in which WildCam failed. By identifying the problems, we identify potential solutions as a guide for the design of a future all-day camera that can validate eating and social presence. Table 2.4 categorizes the reasons for false positives and false negatives for each class label along with their prevalence.

2.6.1 Eating Episodes

We observed two primary causes of false positives in eating episode detection: non-eating-related hand-to-mouth gestures and significant camera motion. Drinking episodes were the main cause of false negatives.

2.6.1.1 Non-eating-related hand-to-mouth gestures—57% of all false positives

One common problem of false positives is detecting hand-to-mouth gestures from gestures not involving food intake. Common techniques such as optical flow often confuse actions like playing with hair, touching the mouth, scratching the face, and putting on headphones or glasses with eating gestures. Prior work has explored the role of food and food detection models in determining eating

moments [82], [83]. Given the current perspective of our camera and its resolution, food detection becomes an even harder challenge. In the future, we can explore removing such false positives by including a chewing detection module that can filter out hand-to-mouth events that do not lead to a chewing action. Alternatively, we could combine a food detection module with a hand-to-mouth detection module separately to prevent such false positives. Figure 2.6a–d shows examples of hand-to-head gestures that are confused with eating.

2.6.1.2 Device movement—43% of all false positives

Detecting an eating episode relies on computing the optical flow. To compute the optical flow, we considered a set of continuous images captured by WildCam and computed the distance by which the points in the image had moved. We observed that, at times, unintended device movement (due to the unstable position on the participant’s shirt) caused motion in the optical flow image that appeared to be similar to hand motion during a feeding gesture. Additionally, in some cases we observed motion artifacts in the camera causing confusion to the model. This highlights the need for simple stabilization strategies such as a firmer mounting clip or inertial sensing to disambiguate device motion from user motion.

2.6.1.3 Drinking gestures—90% of all false negatives

As drinking a beverage contributes toward caloric consumption, we included drinking as an eating activity and annotated the dataset accordingly. However, several prior efforts exclude drinking gestures [54], [79]. Drinking gestures, however, have unique temporal patterns and in our data occurred less frequently than other eating gestures. They also often extended beyond the hand and occluded a large portion of the hand, making automated detection of the hand a challenge. Future work should look into designing a separate beverage detection model to ensure reliable detection

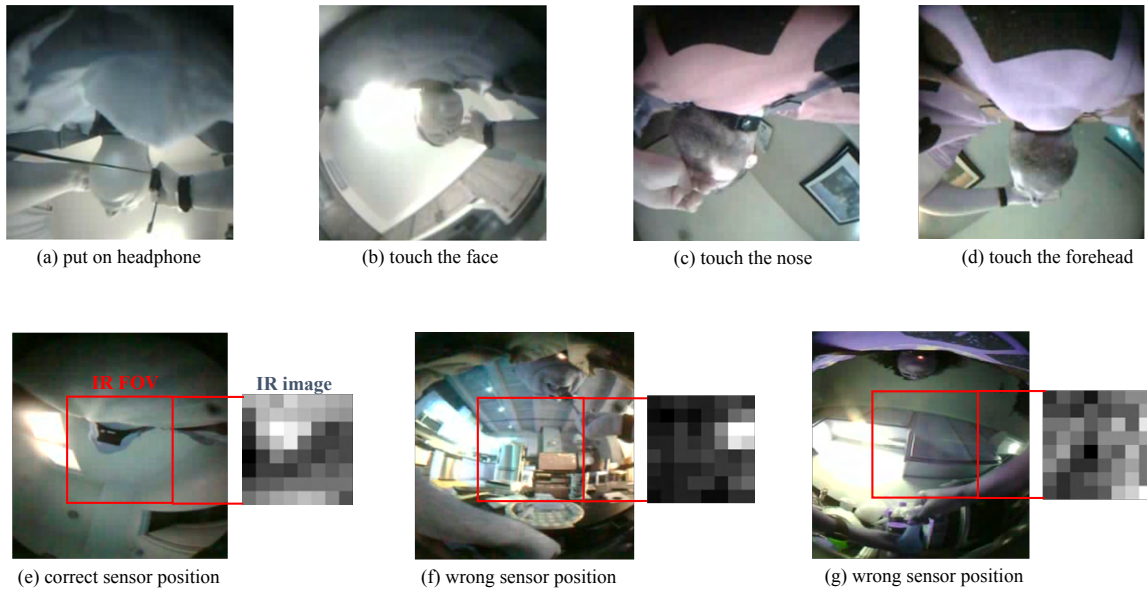


Figure 2.6: Sample cases causing false positives and/or false negatives in the system. a–d: Hand-to-mouth gestures that are confused with eating. e–g: Sensor position can affect the field of view captured by the IR sensor. Red boxes indicate field of view captured by the IR sensor.

of drinking gestures.

2.6.1.4 Partial occlusion—10% of all false negatives

At times the wearer’s clothing or the table would partially occlude the image. Annotators were still able to visually confirm the food in the image; however, the model was not able to recognize the hand in the images. This suggests that exploring other positions for mounting the device and integrating multi-view sensing could help recover eating cues under occlusion.

2.6.2 Social Presence

2.6.3 Social Presence

2.6.3.1 Video calls—18.75% of all false positives

Our current definition of social presence excluded the presence of video calls. However, we observed that several participants spent time in video calls on the smartphone. This highlights the need to either modify the definition to take into account the fact that social presence may occur through video calls or ensure our model is able to distinguish faces detected on screens from faces in real life. Alternatively, we could use data from the person's smartphone to filter out these false positives or combine screen detection with social presence to filter faces bound within screens.

2.6.3.2 I see ghosts—31.75% of all false positives

Due to the low-resolution nature of the RGB camera used, we trained the model using tiny faces with very few pixels on the face. This resulted, at times, to the model "seeing" faces that were not really there. We could potentially maintain a minimum number of pixels on target for detection, or we could slightly increase the resolution of the RGB camera to prevent these false positives.

2.6.3.3 I see myself—30.75% of all false positives

As shown in Figure 2.6f and g, sometimes the participant wore the device incorrectly, with the IR sensor fully covered by the participant's body or with the participant leaning over the IR sensor with his/her face appearing on the other side, resulting in a false positive. To mitigate this problem, a wider IR field of view can allow us to fully detect the participant's body. Integrating a model that can track the participant's face can also help address this problem.

2.6.3.4 *Mirror mirror on the wall—18.75% of all false positives*

Some participants spent a lot of time in front of a mirror doing tasks such as putting on makeup or brushing their teeth. These activities became a part of the individual's daily routines and cannot be ignored. Our face detection model was not able to distinguish between a wearer's face and their face in a mirror, which appeared as a bystander's face. Therefore, we ended up capturing a significant amount of false positives related to this scenario. Needed is a method to detect mirrors to filter out these false positives.

2.6.3.5 *Tiny faces—69.25% of all false negatives*

Given the tiny faces in the image, there were scenarios where the faces were at the edge of the RGB camera. These faces were classified as hard cases according to the WiderFace dataset [75]. The only way to solve this problem would be to increase the RGB camera's field of view or to increase the camera's resolution.

2.7 Discussion and Future Work

Our analysis of RGB and IR data over 3 days from 10 participants with obesity shows that our system can accurately classify eating episodes and social presence in people with obesity in natural settings. Given a clip of RGB and IR sensor data from a camera oriented toward the wearer's mouth, generating images at a low resolution, we address two key technical challenges of *variable orientation* and *cross-sensor field-of-view mismatch*. Variable orientation and resolution refer to the fact that the number of pixels on an object for an activity of interest varies as the participant moves in the scene. For example, given the varying positioning of the device, fish-eye lens, and motion of other people in the scene, the bystander's face can appear at several different resolutions and

orientations. This is in contrast to prior work that assumes higher-resolution images of people, with faces appearing in the same orientation throughout the video footage. We address this by testing our classifiers with images of the object at various resolutions and testing the faces at different rotations to ensure the classifiers can detect tiny faces at different orientations in the scene. The second challenge of cross-sensor field-of-view mismatch has to do with the varying resolution of the IR sensor in relation to the RGB camera, which results in objects appearing entirely in one sensor and partially in another. As shown in Figure 2.1, the IR sensor overlaps with the center of the RGB frame. We handle this mismatch not by relying on the model to implicitly learn it from heterogeneous sensor data alone, but by explicitly using the IR signal only during postprocessing to validate objects that fall within its field of view. In contrast, prior work on object detection using multiple sensing modalities assume similar resolutions [37]. To address this, we used IR sensor data during the postprocessing step of the data pipeline to confirm or refute a detected object in the RGB image only if the object overlapped with the field of view of the IR sensor. Therefore, we could infer the IR sensor’s utility in confirming or refuting the object in the scene. Moreover, we show how future behavioral researchers could use the WildCam system for defining and validating various behavioral constructs.

WildCam currently uses images captured from a wearable camera primarily intended to capture the wearer and activities around the wearer. However, the privacy of both the wearer and bystanders is a concern with camera images. In the future, we plan to only record and train models using obfuscated images. To this end, we will run image obfuscation approaches to remove human faces from the images to further protect bystander privacy [84].

2.8 Summary of Findings

Studying human behavior has traditionally been theory based, deriving a set of constructs or complex concepts (*e.g.*, self-efficacy or attitude toward a behavior) and relationships between variables to explain or predict behaviors. These approaches are often not based on objective, timely measures of behavior but rather on a limited set of tools, implemented as questionnaires administered in structured settings like a classroom or laboratory or over the internet.

Wearable mobile sensors can enable quantitative representation of coarse-grained (*e.g.*, eating alone or with friends) and fine-grained (*e.g.*, number of feeding gestures) behaviors, enabling the development of new behavioral models that more precisely explain and predict problematic behaviors. This will enable behavioral scientists to construct models that more closely represent reality and enable the improved design of interventions that change behavior. In this chapter, we presented how researchers can use a low-power device that consists of an RGB camera and IR sensor to detect eating episodes and social presence via a free-living user study with 80 hours of data. Overall, we showed that the WildCam framework could detect these activities and events reliably when the RGB and IR sensor were used in conjunction. There are several scenarios where the WildCam framework is affected by false positives and false negatives, and we discuss possible modifications to the framework that can help minimize these types of errors. Overall, the WildCam framework will be a powerful tool for behavioral researchers interested in studying human behaviors related to eating and beyond.

CHAPTER 3

ENERGY-EFFICIENT SMOKING DETECTION USING THERMAL IMAGERY

3.1 Problem Statement and Motivation

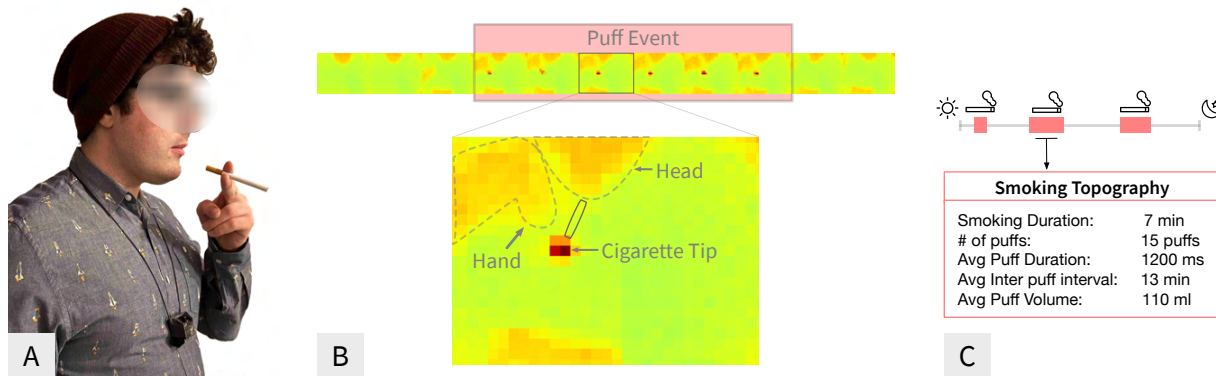


Figure 3.1: SmokeMon is a chest-worn thermal-sensing wearable system with the lens pointed toward the wearer’s mouth [A], capturing spatial, temporal, and thermal information around the wearer and cigarette [B] all day to unobtrusively and passively detect smoking events and estimates of smoking topography [C].

Smoking is the leading cause of preventable death worldwide [85]. Globally, over 8 million deaths are attributed to smoking each year [86]. In the United States, smoking-related illness is responsible for roughly one in five deaths annually [85]. The staggering toll of smoking on human life has spurred large-scale public health efforts to raise awareness of the health risks of smoking and reduce smoking prevalence, including through a US Department of Health and Human Services “Healthy People 2030” initiative.¹ Although smoking is known to be harmful, the majority of what is known regarding the relationship between human exposure to tobacco and the harmful effects is predominantly based on self-report; controlled laboratory environments; environmental sensors;

¹<https://health.gov/healthypeople>

and blood, saliva, or urine tests. These methods are either unreliable and/or expensive or do not capture active smoking behaviors. Exposure to tobacco through smoking still lacks an ideal method of measurement in natural settings [87].

Recent nicotine and tobacco research studies demonstrate the importance of measuring smoking exposure through *smoking topography* or fine-grained measures of smoking patterns. Benefits of studying smoking topography include (1) measuring and assessing harmful carbon monoxide exposure among smokers [88], (2) understanding the relationship between chemical exposure and tobacco related diseases [89], and (3) providing measures to predict abstinence post treatment [90]. Smoking topography is measured during a smoking bout, which provides a summary of fine-grained measures of exposure, including timing of a puff, number of puffs, puff duration, puff volume, inter-puff interval (IPI), and smoking duration. Collectively, these topography measures provide a valid and reliable index of conventional smoking exposure [91].

Gold-standard methods in measuring smoking topography require smokers to attach the cigarette to one end of a device and smoke through a mouthpiece on the other end (*e.g.*, CReSSPocket). The reduced realism of these smoking events alters the way participants smoke, impeding the ability of these devices to capture natural, habituated smoking behavior [92]. To address this, researchers have investigated non-obtrusive ways to measure smoking behavior, including the use of inertial measurement unit (IMU) sensors in smartwatches [93]. However, such approaches are often confounded by non-smoking hand-to-mouth gestures and consequently generate many false positives, suggesting motion-based sensing is inadequate for detecting smoking events and topography accurately when used alone. Wearable cameras, on the other hand, can capture temporal and spatial color information with which smoking can be distinguished from its confounding gestures [94]. However, using video data introduces privacy and stigma concerns, limiting the applicability of camera-based approaches in natural settings [95].

In this chapter, we introduce SmokeMon (Fig. 3.1), an all-day wearable device that uses low-resolution thermal imaging to capture temporal, spatial, and thermal features of wearer smoking activity, including puffing behavior. The *key observation* used in designing SmokeMon is that smoking episodes have visibly distinctive thermal signatures (*i.e.*, the temperature of the tip of a lit cigarette), as well as spatially and temporally distinctive features of hand-to-mouth smoking gestures. Previous work used multiple low-power and low-resolution thermal sensors to classify human activity showing the promise of thermals in classifying smoking activity in a controlled environment [96]. SmokeMon advances current smoking behavior research by using a low-power, low-resolution thermal sensor to not only classify smoking sessions but also extract further information about the smoking activity (*i.e.*, smoking topography) using machine learning in both controlled and natural settings.

The following are the key contributions of this chapter:

1. We present the design and implementation of an all-day wearable system, SmokeMon, along with the data-processing pipeline that enables detection of smoking activity and topography automatically and unobtrusively — that is, without reliance on user input and without physically interfering with the cigarette or smoking gesture.
2. We evaluate SmokeMon’s ability to detect smoking events and extract smoking topography in both controlled and natural settings.
3. We introduce and share the first dataset where smoking topography estimation using thermal imaging is attempted. We release two annotated datasets (in the laboratory and in the wild) to the public, enabling others to expand on the detection of smoking and the estimation of smoking topography using thermal image sensing.

4. We release an open-source SmokeMon data-processing platform to enable further research that aims to understand or measure smoking behavior: <https://github.com/HAbitsLab/SmokeMon>.
5. We report on user feedback to highlight the advantages and the disadvantages of SmokeMon to inform future designs.

3.2 Related Work

3.2.1 Smoking Topography

Smoking topography provides comprehensive quantification of the manner in which someone smokes through variables described in Table 3.1. The ability to quantify smoking behavior at such a level of detail has allowed researchers to further understand factors that influence or maintain nicotine

Table 3.1: Smoking topography measurements and their definitions

Measurement	Definition
A puff	Starts when the person puts the cigarette in their mouth and begins inhaling; ends when the person stops inhaling the smoke. The action of exhaling the smoke is not considered part of the puff.
Number of puffs	Total number of puffs in a smoking session while smoking a single cigarette
Puff duration	During a single puff, the time (in milliseconds) between the start and end time of the puff
Puff volume	During a single puff, the volume (in milliliters) of the smoke inhaled
IPI	The duration (in seconds) between the end of one puff and the start of the next
Smoking duration	The time (in seconds) between the start time of the first puff and the end time of the last puff within a session

use. For example, smoking topography metrics have been shown to predict abstinence following treatment with nicotine replacement therapy, where IPI significantly predicted abstinence at the end of the trial [90]. As a result of this finding, researchers are now paying closer attention to the effect of smoking cessation pharmacotherapy on smoking topography [97]. Smoking topography has also helped explain why cigarettes labeled as "light," "low," or "mild" (i.e., low-yield cigarettes) have failed to minimize the harms of smoking. Herning et al. [98] show that when people smoke low-yield cigarettes, they often take puffs with greater volume and duration than puffs taken while smoking high-yield cigarettes. This suggests that smokers often compensate for the reduced nicotine content of low-yield cigarettes by changing their puffing behavior.

Devices measuring smoking topography have improved significantly throughout their relatively short history. They began as in-lab devices (e.g., Clinical Research Support System [CReSS]) with which users smoke a cigarette through a mouthpiece connected to a pressure transducer to measure airflow, from which smoking topography measures are then calculated [99]. To facilitate smoking topography measurement in natural settings, portable devices have been introduced. CReSS pocket [100] is a portable version of CReSS. Users insert a cigarette into one end and inhale from the other end. Other smoking topography devices have been introduced as well, such as Smoking Puff Analyzer Mobile (SPA-M; SODIM SAS, Fleury-les-Aubrais, France), CReSS-micro (Plowshare Technologies, Inc, Baltimore, MD, United States), and Wireless Personal Use Monitor (wPUM) [101]. Despite their utility, these devices are bulky and obtrusive, requiring the smoker to inhale through a mouthpiece, altering the experience of smoking [99], [102]. Moreover, sharing a smoke-through device between participants requires thoroughly opening and cleaning the device between uses and constitutes a safety risk made especially salient by the ongoing COVID-19 pandemic. These challenges have inspired many researchers to propose alternative approaches to smoking topography, as we discuss in the next section.

3.2.2 Approaches to Detect Smoking Behavior Using Wearable Sensors

Tracking smoking events and topography in natural settings remains challenging. Imtiaz et al. [103] conducted a systematic review of wearables used to monitor smoking behavior in free-living environments, along with the behavioral and physiological manifestations of smoking behavior that drive choices of sensing modalities and approaches in smoking detection. For example, hand-to-mouth gestures are a behavioral manifestation of the puffing event, prompting researchers to use motion-based sensors in wrist-worn devices to track hand-to-mouth gestures as a proxy for smoking behavior. Below we summarize additional behavioral manifestations of smoking events identified by Imtiaz et al. and expand their list to include the thermal manifestation of smoking.

3.2.2.1 *Hand-to-mouth Smoking Gestures:*

Motion-based sensors such as IMUs have been extensively used to track hand-to-mouth gestures as a proxy to smoking behavior. Researchers have used inertial sensors (accelerometer, gyroscope, magnetometer) both individually and in combination to detect these gestures. Parate et al. [104] demonstrate the possibility of detecting the smoking gesture using quaternion information from inertial sensors, whereas Skinner et al. [105] and Shoaib et al. [106] used data from the smartwatch's accelerometer and gyroscope to detect smoking activity. Senyurek et al. [107] used only one axis from a wrist-worn accelerometer to determine the periodic repetition of smoking gestures. Although hand-to-mouth gestures are strong features of smoking behavior and can be used to extract puffing behavior, approaches that rely solely on hand-to-mouth gestures are often confounded by other common daily behaviors that require hand-to-mouth gestures (*e.g.*, eating, drinking, touching the face).

3.2.2.2 *Respiratory Pattern of Smoking:*

Researchers have shown a difference between the pattern of inhalation/exhalation when smoking and the pattern of inhalation/exhalation when breathing regularly. Specifically, Imtiaz et al.'s [103] review paper summarized the respiratory pattern of smoking or puffs into four phases: (1) a short period of breath-holding when the cigarette is in the mouth, (2) a sharp increase of airflow when smoke is inhaled, (3) a brief period to hold the smoke in the lungs, and finally (4) smoke exhalation through the nose or mouth. To measure this pattern, researchers use wearable respiratory inductance plethysmography (RIP) sensors. RIP sensors consist of inductive threads usually embedded in belts or sewn into clothing. During the inhale and exhale process, the lungs contract and expand causing a change in the inductance of the belt[103]. Although such an approach has shown promise in controlled laboratory experiments, in free-living contexts it requires the use of other sensors that capture hand-to-mouth gestures to increase confidence in the detection and eliminate confounding breaths[108]. RIP signals are also known to be similar across multiple activities: standing, walking, resting idly in a chair, eating, and smoking cigarettes [103].

3.2.2.3 *Cigarette Ignition Event*

One way to track the frequency of smoking is by tracking the number of times a cigarette is ignited. Ubi-Lighter [109] and the PACT lighter [110] are examples of smart lighters that can timestamp the cigarette ignition event as a proxy to the start time of a smoking event. Although such an approach does not provide further smoking topography, it is often used with other methods to reduce false-positive events. For example, Senyurek et al. [111] and Imtiaz et al. [110] used a smart lighter in combination with a wrist-based sensor to detect smoking more accurately than could be done with either device alone.

3.2.2.4 Thermal Signature of Lit Cigarettes:

HeatSight [96] explored using multiple low resolutions of 8×8 thermal sensors to detect human activity in a controlled environment. Classification of smoking sessions achieved a very high F-score (83.66%), which shows the promise of using thermal sensors for smoking session classification. In this chapter, we extend previous work and utilize the thermal signature of lit cigarettes to classify smoking sessions and further detect and classify puffs within the smoking session to extract smoking topography. We also collect thermal data from both controlled and natural living environments to provide insights on the feasibility of using thermal as a smoking topography device in the wild. Finally, we provide participant feedback on their experience wearing SmokeMon in the wild, including feedback in collecting thermal data, which can inform future studies that use low-resolution thermal sensors.

Sensing modalities or systems that can capture multiple manifestations or proxies of smoking behaviors have reported higher scores in detecting both smoking events and smoking topography. For example, PACT2 [110] employs a smart lighter in addition to an IMU and RIP to further increase confidence in smoking detection by capturing cigarette lighting events. While multi-sensing approaches improved the scores of smoking detection, they impose an extra burden on users as they require charging and wearing multiple devices. Egocentric RGB cameras have shown significant improvement and potential in measuring many behavioral manifestations of smoking [112]. However, they create privacy concerns, discomfort, or stigma for the wearer and the people around them [113], [114]. *SmokeMon uses a low-resolution thermal camera that can sense many smoking behavior proxies such as the thermal signature of a lit cigarette, hand-to-mouth gestures, some respiratory patterns (e.g., inhalation of smoke causing the temperature tip of the cigarette to increase), and cigarette lighting events, all without capturing unnecessary information in the environment that*

can trigger privacy concerns (as is the case with egocentric RGB cameras).

3.2.3 Thermal Imaging in Human Activity Recognition and Context-Aware Computing

Thermal cameras capture infrared radiation emitted by objects in the environment. Thermal images comprise pixels representing the thermal radiation value of the object in the field of view, which can be used to calculate the temperature of the object. Humans, as well as many objects (including a lit cigarette), naturally emit thermal radiation that can be captured with the thermal sensor unobtrusively (i.e., without the need for contact). Over the years, researchers have used thermal sensing as a *nonwearable* sensing modality for in-room activity monitoring (such as for walking [115]); as an overhead sensor to detect hand interaction with surfaces [116], mid-air gestures [117], and face-fronting (to measure cognitive load [118] or stress [119]); as a hand-held sensor to recognize materials [120]; or to perform thermographic-based energy auditing [121], [122], [123]. As a *wearable* sensing modality, thermal sensors have been used primarily to extract information about the human body [124]. Glimpse [125] used it in a wearable camera to discard frames that have bystanders by detecting humans in the scene. FingerTrak [126] used multiple thermal cameras in a bracelet configuration to predict the wearer’s hand poses. HeatSight [96] used multiple 8×8 thermal sensors to classify human activity, including smoking events, in a controlled setting. However, these sensors do not classify or detect puffing events to extract smoking topography. SmokeMon extends the work of HeatSight by exploring the feasibility of a low-resolution thermal sensor to extract smoking topography. Moreover, we do not use the same thermal sensors used in HeatSight (i.e., Grid-EYE) because it fails to work in sub-zero Celsius climates, ruling out the potential of using the sensor in many seasons and some parts of the world.

3.3 SmokeMon: System Design, Implementation and Energy Consumption

In this section we describe SmokeMon’s design goals along with explorations performed to determine the best design choices, followed by the final system designed to measure smoking topography. We conclude this section by discussing SmokeMon’s energy consumption and battery life.

3.3.1 Design Goals

3.3.1.1 *Goal 1: Reliable sensing of smoking behavior.*

SmokeMon must provide rich information that can capture multiple proxies of smoking behaviors to be more reliable in extracting smoking topography among known confounding activities.

3.3.1.2 *Goal 2: Capture data that humans can interpret.*

Data obtained from SmokeMon should be easily verifiable and interpretable by a human. One of the main challenges in the area of human activity detection is the lack of fine-grained labeled datasets because it is challenging to obtain ground truth in free-living settings. SmokeMon data should address this challenge by capturing data that are intelligible to humans and machines.

3.3.1.3 *Goal 3: Unobtrusive low-burden sensing.*

SmokeMon should minimally interfere with everyday activity, especially smoking. Wearable devices can introduce stigma and privacy concerns. The design, sensor position, and data collected from SmokeMon should not introduce additional burden on the wearer, affecting them neither physically nor psychologically.

3.3.1.4 Goal 4: Low-power sensing to support all-day wear.

Smoking can happen at any time of the day. Therefore SmokeMon should last for at least 16 hours on a single charge. This number is calculated based on waking hours, assuming that a person needs 8 hours of sleep daily. It is also essential that SmokeMon incorporates energy-efficient approaches and components to reduce the weight and size of the battery while still providing all-day sensing.

3.3.1.5 Goal 5: Open platform to support smoking cessation studies.

Smoking topography research has shown great promise in understanding smoking and nicotine intake habits and assessing the impact of smoking cessation interventions. SmokeMon should provide an open platform to the research community to continue advancing the unobtrusive capture of longitudinal information about smoking behavior.

3.3.2 Design Exploration: Motivations and Challenges in Using Thermal for Smoking Detection

To attain the aforementioned design goals, we explored multiple approaches. The thermal-based sensing approach provided the greatest promise in attaining the stated goals. We describe the advantages of using a thermal-based approach in detecting smoking behavior.

3.3.2.1 Motivation 1: Low-resolution thermal camera for comfortable all-day sensing.

Low-resolution thermal cameras are light weight and energy efficient, making it possible to wear them continuously without burdening the participant. Because the thermal camera is worn on-body, the person will be in close proximity to the camera, making it sufficient to use a low-resolution thermal camera. However, it is necessary to ensure that the field of view (FoV) of the thermal

camera is capable of capturing the wearer's hand and head movement.

Table 3.2 provides metrics and comparisons between thermal cameras available off-the-shelf. We specifically compare three different cameras that have different characteristics: (1) FLIR ² is a high-resolution thermal camera, (2) MLX ³ is a low-resolution thermal camera, and (3) Grid-EYE ⁴ is an ultra-low-resolution thermal camera. As evident from the sample images presented in Table 3.2, each device is capable of capturing a silhouette of the human head when the camera is pointed upward from the chest, regardless of the device's resolution. However, we note that FLIR has an extremely high current draw compared with the other devices, which makes it unsuitable for longitudinal data collection (affecting our previously stated Goal 4). We thus eliminated FLIR from our consideration. Initially, we were interested in using the Grid-EYE sensor (similar to the one used in HeatSight [96]) as it has a low current draw. To ensure that Grid-EYE addressed our needs, we conducted a small experiment with the Grid-EYE device in which we asked participants to smoke while wearing the device. However, Grid-EYE failed to work in contexts with an ambient temperature below $0^{\circ}C$ and therefore restricted the use of the Grid-EYE sensor to a few months and limited regions of the world (failing Goal 1 and Goal 5). We also noticed that when the participants smoked sideways, the cigarette would go out of the FoV. Therefore, we opted for the MLX sensor due to its larger FoV, allowing for a more robust capture of cigarette motion under many ambient temperature settings. Additionally, the larger FoV made it less sensitive to wearing position.

One major advantage of thermal cameras is that they can capture thermal images even when they are covered with opaque materials. The ability to conceal the thermal camera increases the participant's comfort in wearing them. Researchers have shown that people grow uncomfortable

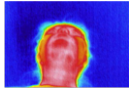
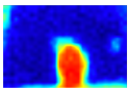
²FLIR 3.5 <https://www.flir.com/products/lepton/?model=500-0771-01&vertical=lwir&segment=oem> (Accessed June 5, 2022)

³MLX 90640 <https://www.melexis.com/en/documents/documentation/datasheets/datasheet-mlx90640> (Accessed June 5, 2022)

⁴Grid-EYE <https://industry.panasonic.eu/products/components/sensors/ir-thermophile-array-sensor-grid-eye> (Accessed June 5, 2022)

when they see a camera lens, which in turn makes the wearer feel stigmatized, resulting in social avoidance, even if the camera does not record the bystander [113]. Therefore, to increase comfort in wearing SmokeMon, we investigated different materials to cover the MLX lens. We tried a thermal-passing acrylic sheet and a thermal-passing black thin plastic sheet (similar to a garbage bag). We chose the thin plastic sheet because it was easier to wrap around the opening of the camera.

Table 3.2: Comparison between thermal cameras available off the shelf

	FLIR	MLX	Grid-EYE
Resolution (p)	160×120	32×24	8×8
FoV ($^\circ$)	57	110×75	60×60
Temperature range ($^\circ C$)	-1 to 400	-40 to 300	0 to 80
Sensitivity ($^\circ C$)	0.05	0.1	0.16
Current (mA)	45.5	23	4.5
Per Unit Cost (USD)	233.99	74.95	45.95
Sample Image			

3.3.2.2 Motivation 2: Thermal sensors provide rich information to model smoking topography.

It is well understood that the more data we can capture from diverse smoking instances, the better SmokeMon’s ability to predict and correctly model smoking behavior. Several observations confirmed that the thermal sensor provides rich smoking-related information. SmokeMon takes advantage of spatiotemporal thermal data to capture the moment when someone lights up the cigarette. Also, the temperature of the cigarette during a smoking session provides valuable information about the puff. Figure 3.2 shows a time series signal of the maximum pixel temperature during a smoking session. We also highlight the ground truth of a puff in the same figure. One must note that thermal information alone can not predict puffs accurately. We can see this in Fig. 3.2C, where

the cigarette is in the FoV but not in the mouth. We see a peak in the maximum pixel temperature when the cigarette is in the FoV. However, the spatiotemporal thermal information clearly shows that the cigarette is not near the mouth. Moreover, during the puff, we can see an increase in the cigarette temperature caused by the combustion process when the participant inhales air through the cigarette. This information can be used to approximate puff volume. Fig. 3.3 shows the correlation (Pearson's $r=0.77$) between the normalized ground truth puff volume obtained from CReSS Pocket and the one calculated from the thermal data by integrating over the maximum temperature rate of change during the puff.

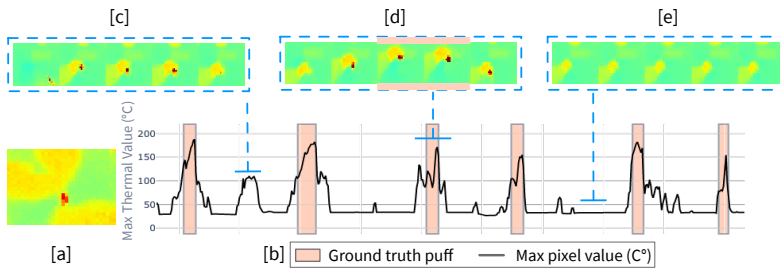


Figure 3.2: Thermal captures fine-grained information about the smoking session including: [A] cigarette ignition and [B] cigarette tip during a smoking session. Thermal, spatial, and temporal information can help in distinguishing between a non-puff (when the cigarette is in the FoV but not in the mouth [C]) and a puff (when the cigarette is near the mouth [D] or when there is no cigarette [E]).

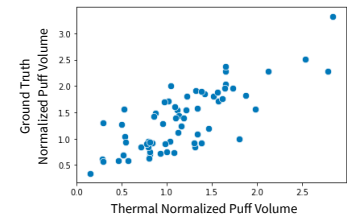


Figure 3.3: Correlation (Pearson's $r=0.77$; $p<.01$) between the normalized ground truth puff volume obtained from CReSS Pocket and the one calculated from the thermal data.

3.3.2.3 Motivation 3: Thermospatial data are useful in distinguishing confounding hand-to-mouth gestures.

Hand-to-mouth gestures are often confounding; research has shown how motion-based sensors like IMU often confound similar hand-to-mouth gestures like eating, drinking, and smoking [127]. Thermal images provide useful information to the human and the machine, which helps in distinguishing between similar and often confounding hand-to-mouth gestures. Fig. 3.4 shows examples

of different hand-to-mouth gestures with and without objects in hand, such as smoking, eating, drinking, and touching the head or mouth. We use a color map that maximizes the viewability of the image to a human reviewer. From the images, we can observe that the smoking gesture has a clear signature that is easily observable by a human reviewer. As we will show in subsequent sections, a machine can also learn to capture this signature.

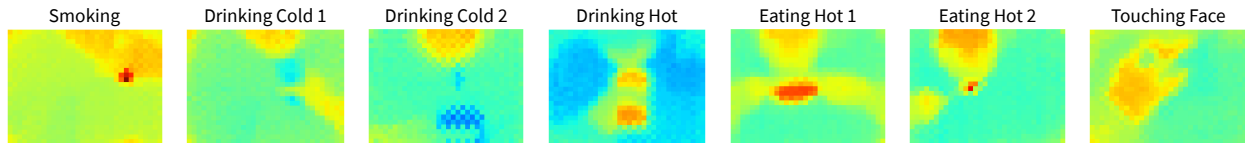


Figure 3.4: Thermal sensors capture rich information that can help distinguish between many hand-to-mouth gestures.

3.3.3 SmokeMon Hardware, Firmware, and Encapsulation

3.3.3.1 Hardware:

For the development of the SmokeMon prototype, we used a development board ⁵ equipped with a Cortex-M4-based Apollo 3 microcontroller and a BLE 5.0 radio. In addition, the board also contains an MCP73831 single-cell LiPo charger, a real-time Clock (RTC), and a micro-SD card socket, which are essential components for in-wild deployment. BLE and the micro-SD card slot are required for communication and data storage. SmokeMon obtains the thermal images from a breakout board that contains an MLX90604 thermal sensor array ⁶ and connects to the main board via an I2C-based connector. The size of each captured thermal image is 32×24 pixels, where each pixel represents a temperature reading ranging between -40°C to 300°C .

⁵Sparkfun OpenLog Artemis <https://www.sparkfun.com/products/16832> (Accessed August 15, 2022)

⁶Adafruit MLX90640 <https://www.adafruit.com/product/4469> (Accessed August 15, 2022)

3.3.3.2 Encapsulation and Attachments:

It is crucial to encapsulate SmokeMon in a wearable case that maximizes comfort and allows reliable data collection. The thermal camera in SmokeMon should be positioned in a way that allows capturing of the head and as much of the hand trajectory as possible. This allows confirmation of the cigarette going toward the mouth. Therefore, we printed multiple mounts with different angles (centering the camera toward the chin, at 45° to 60° from the chest) to hold the thermal sensor based on the wearer's chest size (see Fig. 3.5). For example, a person with a flat chest will have a mount with a 55° or more, whereas a person with a protruding chest will require a smaller angle. We also fixed the position of the device on the chest using a necklace chain of fixed length. We determine the best angle and chain length for the participants during the laboratory visit; we confirm the angle and length by visualizing the data obtained from the SmokeMon device. To prevent SmokeMon from flipping or moving out of place, we provide a magnet attachment mechanism for participants to place a magnet behind their clothing to connect with the magnets with opposing poles embedded behind the SmokeMon case.

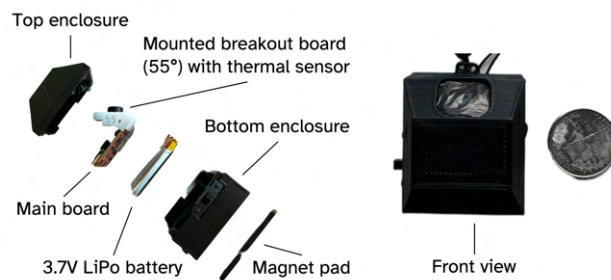


Figure 3.5: Enclosure and attachment mechanisms in SmokeMon to maximize comfort while enabling reliable data collection.

3.3.3.3 *Firmware:*

SmokeMon firmware samples a timestamped thermal frame at 4 Hz (equivalent to 4 frames per second [fps]). Depending on the objective for using SmokeMon, the firmware can be set to save all timestamp data to an external SD card (*Storage State*) or can be set to stream data to another device using BLE (*BLE state*). The *storage state* is useful when continuous and reliable data is favored over real-time streaming or processing of the data. For example, in our case, we would rather save all collected data on an SD card to increase the reliability and validity of our experiment. Similarly, smoking topography devices (*e.g.*, CReSS Pocket) perform analysis offline at the end of the data collection process. On the other hand, a smoking cessation application might utilize *BLE state* to send notifications to the user as a form of intervention or might need to offload data for further real-time processing on smartphones. SmokeMon facilitates both cases by enabling the researchers or users of SmokeMon to choose the firmware state based on their goals.

3.3.4 Energy Consumption and Battery Life

3.3.4.1 *Energy Consumption:*

Table 3.3 presents the average current draw of each state that SmokeMon supports, including the *storage state* and the *BLE state* explained above. In addition to these two states, SmokeMon also supports the *sensing-only* state. We use the INA219 current sensor ⁷ to measure the power consumption of SmokeMon in each firmware configuration: *sensing only*, *storage* and *BLE* states. To measure the current, we connected the current sensor between the power source (500 mAh LiPo battery) and SmokeMon GND and VIN contacts on the board. We then connected an Arduino board to the same current sensor via I2C to read the power profile of the system at a sampling frequency

⁷INA219 <https://www.adafruit.com/product/904> (Accessed June 5, 2022)

of 50 Hz. For each state, we collected the power profile data for 1 hour and calculated the median current consumption for each state, as reported in Table 3.3. We used these numbers to estimate the battery life of SmokeMon.

Table 3.3: Current draw and power consumption of SmokeMon firmware under different settings: (1) sensing-only state: collects and processes the thermal data; (2) storage state: collects, processes, and stores the data on an SD Card; and (3) BLE state: collects, processes, and streams the thermal data over BLE.

	SmokeMon Firmware		
	<i>Sensing Only</i>	<i>Storage</i>	<i>BLE</i>
Current Draw (mA)	24.2	26.1	33.6
Power (mW)	91.9	104.4	132.7

3.3.4.2 Battery Life:

For our data collection, we used SmokeMon’s *storage* state firmware. Based on the estimated current consumption for SmokeMon *storage* state firmware shown in Table 3.3 and based on the fact that we want the system to last for a full waking day (at least 16 hours, assuming 8 hours of sleep), we chose to equip SmokeMon with a 500-mAh battery, resulting in a battery lifetime of 19 hours. To ensure that our estimation is correct, we fully charged SmokeMon and asked a volunteer to wear the device until the battery drained completely. We instructed the volunteer not to switch off the system but rather take it off and place it on a table at the end of the study. When the battery was fully depleted, we estimated the battery lifetime by calculating the difference between the first and last timestamps of readings stored on the SD card.

Values in Table 3.3 can also be used to estimate battery lifetime when multiple states are used. For example, instead of continuously saving or streaming data, a more energy-efficient approach will only save or send the data when there is a high probability of a smoking event (*i.e.*, when a very hot pixel is detected). In Section 3.6.3, we perform such analysis based on the data that

we collected to demonstrate how an opportunistic data collection approach can further reduce the power consumption of the system to enable longer battery life or to facilitate real-time processing via BLE.

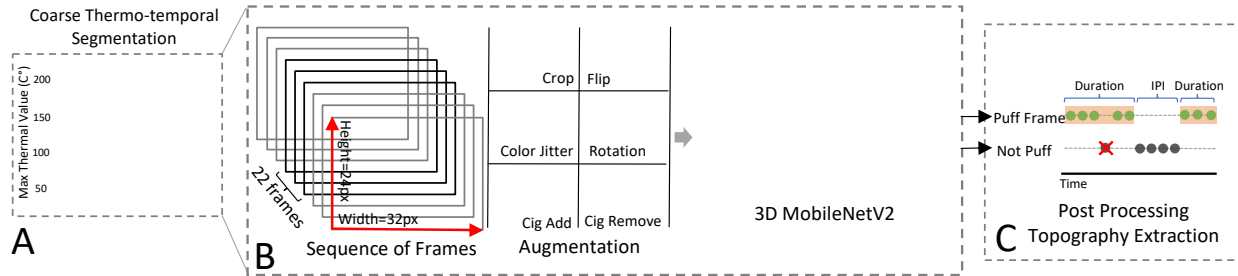


Figure 3.6: SmokeMon framework to extract smoking topography. [A] We first extract possible smoking sessions based on thermal information. Then we [B] run a puff-detection model on each frame in the extracted session. Finally, [C] we perform postprocessing to extract puffing events and the smoking topography of the session.

3.4 SmokeMon Framework to Extract Smoking Topography

Figure 3.6 illustrates the SmokeMon framework for extracting smoking topography. SmokeMon first performs coarse thermo-temporal segmentation to automatically discard time segments with low probability of smoking. Within each remaining time segment, SmokeMon runs a puff classification model on each frame and then constructs a puff event by post-processing the frame-level puff prediction. Finally, SmokeMon extracts the smoking topography based on the constructed puff events.

3.4.1 Step A: Coarse Thermo-Temporal Segmentation

Passive wearable systems deployed in free-living settings collect massive amounts of data about both relevant (*i.e.*, smoking) and irrelevant activities. Running a machine learning model on each

time segment is inefficient as it unnecessarily increases computational load and energy consumption. Since we are interested in smoking segments of the data, we perform a coarse thermo-temporal segmentation to discard irrelevant time segments and keep time segments with a high probability of involving smoking activity (*i.e.*, segments that have at least one pixel with a high temperature value that may belong to the tip of a burning cigarette). To reduce the runtime and energy consumption of the coarse thermo-temporal segmentation step, we designed a heuristic-based algorithm (see Algorithm 1) that can efficiently run either on-device on the SmokeMon MCU or on an offline CPU after extracting the data from the SD card. The algorithm loops over the frames and checks the maximum pixel value of each frame F_i and creates a candidate smoking session if the maximum value is $> 70^\circ$ (determined empirically based on the lower bound of the cigarette tips in our dataset). For each F_i frame that is flagged as containing potential smoking activity, the algorithm also flags previous and future frames within the range of $[F_{i-start_offset}, F_{i+end_offset}]$ to include likely puff events without cigarette visibility and obtain a coarse thermo-temporal segment. The offset is a variable in the algorithm that can be adjusted based on the IPI duration. The resulting segments include both smoking and other confounding contexts or activities that can have a high thermal pixel value (*e.g.*, hot drinks and spotlight lighting). In the next step, we use a deep learning approach to classify puff frames to further eliminate confounding and non-smoking frames.

3.4.2 Step B: Puff Frame Detection

We train a deep neural network model to determine whether a participant was performing a cigarette puff in frame F_i . For each frame F_i , frames $F_i - 11$ to F_{i+11} are stacked to create a 3D input data of size $23 \times 32 \times 24$, where the first dimension (23) corresponds to the temporal sequence of frames. This input is then fed into the neural network. We use the MobileNetV2 architecture [128] because it is a light model that can run efficiently on edge devices and enable real-time inference

Algorithm 1 Coarse Thermo-Temporal Segmentation

Require: start_offset = 1320; end_offset = 2640; temperature_threshold = 70

```

1: historical_buffer = [ ]           ▷ The buffer to hold the latest frames (size=start_offset)
2: segment_end = 0
3: smoking_sessions = { }
4: while True do
5:   time, data ← getFrame()
6:   historical_buffer.update((time,data))
7:   if time ≤ segment_end then           ▷ session segmentation continues
8:     smoking_sessions[time] ← data       ▷ flag the time instance as smoking
9:   else if MaxTemp(data) ≥ temperature_threshold then           ▷ check if cigarette exist
10:    segment_end ← time + end_offset       ▷ initiate a segment
11:    for time, data in historical_buffer do           ▷ flag previous time instances as smoking
12:      smoking_sessions[time] ← data
13:    end for
14:  end if
15: end while

```

if needed. We replace the 2D convolution layers with their 3D counterparts to capture temporal information related to the puff gesture. Instead of randomly initializing the weights in our neural network, we use transfer learning and initialize our network with weights obtained from a similar model pretrained on a gesture-recognition dataset [129].

To overcome the challenge of between- and within-subject variability of smoking gesture, we augmented our training set by using the following transformations (see Fig. 3.6 for visuals): (1) We randomly remove cigarettes in the frame to force the neural network to learn puffing gestures with and without a visible cigarette. We accomplish this by first locating the cigarette in the frame and then using an in-painting method to remove the cigarette. (2) We also randomly add cigarette confounding blobs in the image to ensure that the network learns the difference between a cigarette in hand and a random hot blob that looks like a cigarette. (3) Since temperature readings are affected by environmental factors such as cold weather, we arbitrarily applied brightness jitter to images, a common data-augmentation technique. (4) We applied other common augmentation methods such

as random crop, random flip, and random rotation to account for device position variability. To further increase the resilience of our models, we intentionally collect data from two participants performing confounding gestures such as drinking water of varying temperature and touching their face (one hour each person) to augment the training set.

We used binary cross entropy loss function with Adam optimizer and chose a learning rate scheduler to run gradient descent until convergence. The model’s performance was measured with a leave-one-participant-out cross-validation method. In this chapter, we do not perform any on-device machine learning prediction; however, we made efficient choices about model architecture to enable future on-device processing.

3.4.3 Step C: Post-processing and Smoking Topography Extraction

After detecting the smoking frames, we cluster the puff frames based on temporal proximity (12 seconds maximum) to create a puffing event. Clusters with less than 2 positive puffing frames are discarded. After obtaining the puff events, we calculate the following smoking topography: number of puffs, puff duration, and IPI. For the in-lab data only, we also calculate the puff volume and use a smoking topography device (CReSS Pocket) for ground truth. Puff volume is calculated from thermal data by first normalizing the data using each participant’s mean and standard deviation. Then for each puff event, we extract a time series of the maximum value of each frame in the puff event to create a maximum temperature curve. The change of maximum temperature within a puff is highly correlated with airflow going through the cigarette and measured by CReSS Pocket. To measure volume, CReSS Pocket generates a flow curve (a time series measurement of flow generated using a pressure transducer) for each puff and integrates over the flow curve. In SmokeMon, we integrate over the max temperature curve within the puff to estimate volume. To obtain a mapping between our estimation and ground truth, we train a linear regression model using leave-one-participant-out

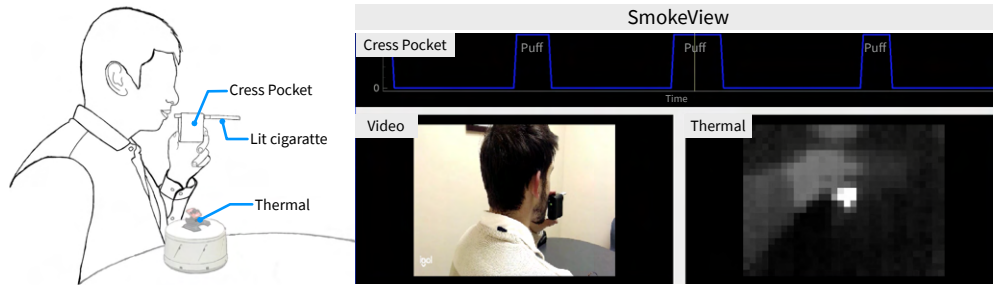


Figure 3.7: In-lab experimental setup.

cross-validation to estimate the puff volume.

3.5 Experiments and Datasets

To evaluate SmokeMon, we conducted an in-lab and in-wild experiment that we further explain below.

3.5.1 In-Lab Experiment

We conducted a controlled in-lab experiment to compare the performance of SmokeMon against the research-grade smoking topography measurement device CReSS Pocket. The elements of smoking topography we were interested in for the in-lab evaluation were the number of puffs, puff duration, puff volume, and the duration of the inter puff interval.

3.5.1.1 Smoking Topography Ground Truth

To extract ground truth smoking topography, we use CReSS Pocket, a popular portable device used in clinical settings. CReSS Pocket is a battery-powered device that automatically measures smoking behavior parameters such as date, time, start and end of the smoking episode, puffs per cigarette, puff volume, and puff duration. In our evaluation, we compared the smoking topography

extracted from our pipeline with the smoking topography obtained from CReSS Pocket.

3.5.1.2 Protocol

Participants were asked to smoke a cigarette using CReSS Pocket and SmokeMon together to provide smoking topography measurements from both devices for each smoking episode. To use CReSS Pocket, participants inserted a lit cigarette into one end of the device and inhaled through the other end using a disposable (See Fig. 3.7). During the pilot study, we noticed that the unnatural position of the cigarette caused by using CReSS Pocket obstructed SmokeMon’s view of the cigarette. In light of this, the experiment was revised so that SmokeMon was not worn but rather placed on the table in front of the participant to compensate for the cigarette offset created by CReSS Pocket (see Fig. 3.7). Because the timekeeping mechanisms of the two systems differ, all sessions were videotaped to facilitate synchronization of SmokeMon and CReSS Pocket data. After teaching participants how to smoke using CReSS Pocket, we left the participant alone in the room and asked them to smoke a cigarette. Participants smoked one cigarette of their choice (provided by the study team) and were compensated \$10 for their time. This experiment was conducted before the COVID-19 pandemic.

3.5.1.3 Dataset

We collected data from 8 participants (7 male, 1 female). All participants (except one) were students at Northwestern University. Details about the collected smoking topography using CReSS Pocket can be found in Table 3.4. To synchronize data collected from SmokeMon, we developed SmokeView to visualize the data we collected during the experiment: video, SmokeMon thermal data, and CReSS Pocket puff time series (see Fig. 3.7). We marked the lighting event, the first puff in the video, and the first puff observed in SmokeMon to calculate the offset between the video

Table 3.4: Puffing statistics obtained from CReSS Pocket

P	Average Puff Duration (s)	Average IPI (s)	Average Puff Volume (ml)	Smoking Duration (m)	Number of Puffs
P1	2.62	16.41	170.06	4.00	15
P2	2.36	17.46	207.70	6.55	21
P3	1.63	11.80	91.02	5.71	22
P4	2.53	13.63	102.92	6.65	26
P5	3.90	17.77	274.89	8.27	23
P6	4.45	11.57	233.48	4.59	38
P7	2.91	8.20	124.22	4.20	22
P8	1.91	9.72	129.05	2.84	11

and SmokeMon. We also marked the first puff using CReSS Pocket to synchronize CReSS Pocket data to the video. After syncing the data, we removed the puffs that were not captured by CReSS Pocket, which were puffs used to light the cigarette. The annotated thermal data and CReSS Pocket data will be released to the public.

3.5.2 In-Wild Experiment

The objective of the in-wild experiment was to test SmokeMon under realistic everyday conditions. We asked participants to wear the device in a natural setting for at least 10 hours (see Section 3.5.2.1). Participants were asked to keep a log of their smoking session, which we then used to obtain a fine-grained annotation of the smoking session and its puffs (Section 3.5.2.3). Table 3.5 provides summary statistics of the data collected in the wild, which we discuss in Section 3.5.2.4

3.5.2.1 In-Wild Protocol

We recruited participants using ResearchMatch.org, Craigslist.com, and flyers distributed around the campus located in a metropolitan city [anonymized for review]. Participants were instructed on how to wear and operate the device. They were asked to wear the device in the laboratory and

smoke and drink a beverage of their choice in the laboratory to ensure that they understood the instruction and to answer any further questions they might have. Participants also had a chance to see a sample of the thermal images collected in the laboratory to demonstrate the type of information that we were collecting as part of the informed consent process. We asked the participants to wear SmokeMon for at least 10 hours in the wild and requested that it include smoking and non-smoking events (*i.e.*, eating, drinking, etc.). Participants were also asked to log the start time of their smoking event by noting it in an electronic or paper log. All participants were compensated US\$60 and were given a chance to review and delete data if requested. No participant deleted any data. At the end of the experiment, we asked participants to fill out a post-experiment questionnaire.

3.5.2.2 *Post-Experiment Questionnaire*

At the end of the study, participants engaged in a post-experiment questionnaire. We asked them about their general impression of the experience and whether they faced any discomfort while wearing the device or if it interfered with their daily lives. We asked participants to report bystander reactions and any discomfort communicated to them by the bystanders. We also ask participants questions about their thermal data and if they had any concerns about sharing it.

General feedback questions. To gain a general impression of SmokeMon, we asked participants the following two open-ended questions: [Q1] *"What did you think about wearing the device to collect data about your activities?"* [Q2] *"What can we do to improve the experience of wearing the device?"*

Participants comfort questions. To understand the user concerns that SmokeMon can illustrate, we asked participants the following open-ended question: [Q3] *"Were there any situations when wearing the device that made you feel uncomfortable?"* [Q4] *"Did you ever take off or cover the device? If yes, when and why?"* We also asked the participants if the device intervened with ev-

eryday activity as a proxy to measure possible user burden and concerns. We specifically asked: [Q5] *"Did wearing the device change how you normally go about your daily activities?... how you smoked? (if yes, how?)"*

Bystander reaction questions. Imaging sensors often cause bystander concerns. However, it is unknown how bystanders will react to the thermal sensor in SmokeMon. To capture bystander reaction, we asked participants: [Q6] *"Did anyone ask you about the devices? If yes, what did you tell them?"* and [Q7] *"Did anyone ask you to take the device off or turn it off/around?"*

Attitude toward data sharing questions. To understand potential data sharing concerns, we asked participants to review part of their data and delete any segments they did not want to share with the researchers. We then asked them [Q8] *"Have you deleted any part of the data you collected? Why?"* We also asked them to report their concern level if we released their data to multiple stakeholders to understand the potential privacy concerns of thermal data produced by SmokeMon. Specifically, we asked them [Q9] *"To what extent would it concern you if we released the data you just viewed to each of the following: other researchers, members of your family, your friends, your doctor, your dietitian, and finally the general public?"*

3.5.2.3 In-Wild Smoking Topography Ground Truth

We labeled the start and end of each puff gesture in a smoking event, and we used participants' self annotation of their smoking to locate the smoking events. We also viewed the entire dataset if a participant reported that they forgot to log a smoking event. We then calculated the following topography from our ground truth labels for each smoking event: (1) start and end of smoking event, (2) puffs per cigarette, (3) puff duration, and (4) IPI. Puff volume could not be extracted from the in-wild data because we did not have ground truth for it. For the in-lab experiment, we used CReSS Pocket for ground truth. However, we could not use CReSS Pocket as a ground truth

device in the wild because it interferes with smoking behavior. Moreover, sharing an inhale through a device during the recent and ongoing COVID-19 pandemic would impose unnecessary risk on the participants.

3.5.2.4 In-Wild Dataset

We collected a total of 110 hours of data containing 115 smoking sessions obtained from 11 participants (different participants than in the laboratory). Table 3.5 shows the number of smoking and puff event statistics for each participant. The deidentified, annotated dataset of participants who consented to release their data will be shared publicly.

Table 3.5: In-wild participants self-reported demographics and number of cigarettes smoked per day and in-wild participants smoking data and topography summary

P	Age	Gender	Cigarettes Per Day	Number of Sessions	Average Smoking Duration (m)	Number of Puffs (s)	Average Puff Duration (s)	Average IPI (s)
P9	23	Non-Binary	2	1	4.22	12	1.64	21.25
P10	56	Woman	12	12	6.28	198	4.11	27.88
P11	75	Woman	20	14	5.11	185	2.75	22.09
P12	24	Man	3	11	6.52	205	2.48	22.54
P13	32	Woman	2	6	3.95	48	1.58	32.03
P14	39	Woman	25	19	5.94	172	1.46	45.78
P15	45	Woman	10	10	2.26	33	1.71	33.20
P16	42	Woman	10	15	4.18	143	3.56	29.85
P17	39	Man	3	11	3.94	81	2.57	39.82
P18	33	Woman	13	15	7.41	146	3.43	54.68
P19	24	Man	2	2	5.70	20	1.56	36.48

3.5.3 Evaluation Criteria and Metrics

We evaluated our smoking topography extraction approach using a leave-one-participant-out method in which we tested the models on one participant and trained the models with the remaining participants. We report our performance in two tasks: puff detection and smoking topography extraction.

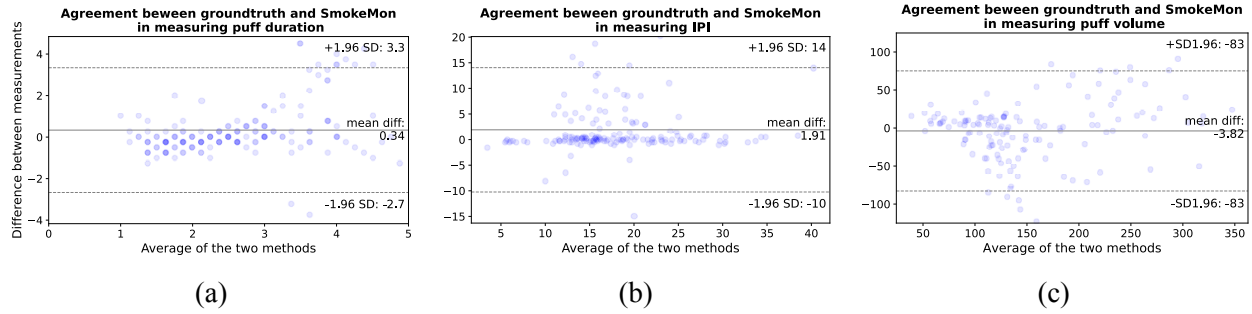


Figure 3.8: In-lab Bland–Altman plots comparing ground truth and SmokeMon smoking topography measurements.

Our model predicts puffing frames followed by a post-processing method that creates events and is evaluated at an event level. We report metrics such as the positive precision, recall, F1 score (positive), and intersection over union (IoU) between ground truth and predicted puff events for each participant individually. Using the true positive puff events, we report the error in our smoking topography measurements. We only use the true positive puffs to prevent carryover errors from the detection task, which we report independently. To analyze the agreement between SmokeMon measurement and ground truth, we used Bland-Altman plots, a method often used to compare gold standards and new instruments in medical and clinical research.

3.6 Results

3.6.1 In Lab: Puff Detection and Smoking Topography

Table 3.6 shows the result of puff event detection from our in-lab experiment using SmokeMon. The average positive F1 score for puff detection was high (0.9). The per-person F1 score was also high (above 0.85) for most of the participants. We inspected low puff detection results to understand the source of the error and the limits of our current model. For P4, recall was very low compared with other participants. Upon inspecting P4’s false positives, it appeared that P4 kept the cigarette

Table 3.6: In-lab puff detection and puff count per participant

P	Puff Events Detection				Puff Count	
	Recall	Precision	F1	mIoU	Ground truth	Predicted
P1	0.93	0.82	0.87	0.69	15	17
P2	0.95	1.00	0.98	0.78	21	20
P3	0.91	0.87	0.89	0.76	22	23
P4	0.96	0.60	0.74	0.71	26	42
P5	1.00	0.92	0.96	0.80	23	25
P6	1.00	0.83	0.90	0.55	38	46
P7	1.00	0.96	0.98	0.79	22	23
P8	0.82	0.90	0.86	0.72	11	10
mean	0.95	0.86	0.90	0.73	-	-

near the face outside of puff events, and our model predicted this action as a puff. Augmenting the training dataset with such cases can further improve the model prediction. Overall, SmokeMon achieved 0.95 recall, 0.86 precision, and 0.90 F1 score in the in-lab experiment, with more than 50% mean IoU.

Fig. 3.8 shows the Bland–Altman plots that compare smoking topography obtained from ground truth and those obtained from SmokeMon. The mean difference was minimal in all plots (0.34 sec for puff duration, 1.91 seconds for IPI, and -3.82 for puff volume). For puff duration and IPI, most points were close to the zero line; the closer the points to the zero-line, the greater the agreement between ground truth and SmokeMon measurements. The puff volume plot shows greater variability, particularly as the volume average increases, indicating a systematic bias in our measurement or the calibration process of our ground truth device CReSS Pocket. However, most points were between ± 1.96 standard deviations (SDs), which was considered acceptable. Among the outlier points (*i.e.*, located outside of the ± 1.96 SD limit) for each plot, all puffs that were above +1.96 SD belonged to P6 with long duration puffs (mean = 6.1 sec). These long duration puffs occurred when the participant kept the cigarette in their mouth while taking multiple puffs.

3.6.2 In Wild: Puff Detection and Smoking Topography

Table 3.7 highlights the results of puff event and smoking session detection from the in-wild experiment using SmokeMon. We extracted all smoking sessions (Recall 1, in Table 3.7). The model average F1 score was 0.8 across all participants and above 0.75 for most participants. We investigated P15’s false-positive and false-negative cases because the recall and precision were very low. It appeared that the position of the device for P15 was quite different than other participants (*i.e.*, head not fully visible in the FoV), which accounted for low recall. Also, P15 kept the cigarette near their mouth and visible in the FoV between puffs, which confused the model with false positives. Overall, SmokeMon achieved 0.82 recall, 0.78 precision, and 0.8 F1 scores for the in-wild experiment with more than %50 mean IoU.

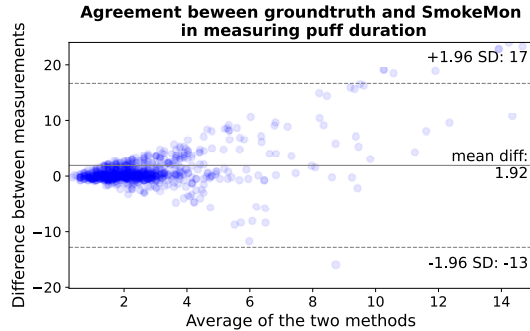
Fig. 3.9 shows the in-wild Bland–Altman plots that compare smoking topography obtained from ground truth with the ones obtained from SmokeMon. The mean difference was minimal in all plots (1.92 sec for puff duration and -3.33 sec for IPI). In both plots, most points clustered around the zero-line, showing high similarity between the ground truth measure and SmokeMon. Puff duration and IPI error increase and fluctuate more frequently as the average increases. However, similar to the in-lab results, most points were within the ± 1.96 SDs considered acceptable. Among the outlier points (*i.e.*, located outside ± 1.96 SD limit) for each plot, all puffs that were above +1.96 SD belonged to a few sessions from P10 with long duration puffs (mean = 5.4 sec).

3.6.3 Energy Consumption Analysis

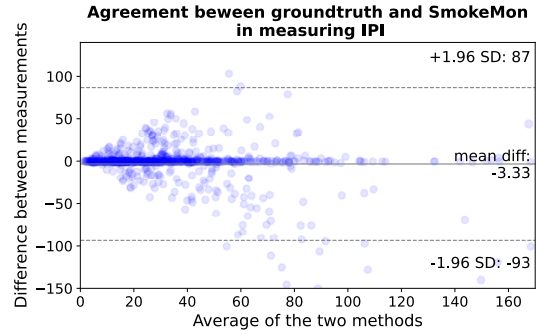
In Section 3.3.4, we profiled each state supported by the SmokeMon firmware, including sensing only, storage, and BLE. For our data collection, we set the system to operate in the storage state to increase our experiment’s reliability and validity. Using the storage state continuously still produced an energy-efficient system; we achieved 19 hours of battery life. In this section, we further

Table 3.7: In-wild average puff prediction per participant, puff count, and smoking session detection. P10 smoking session detection analysis was removed because the ratio of smoking events to nonsmoking events was not similar to other participants because the participant did not follow the study protocol and turned the device on during smoking sessions only.

P	Puff Events Detection				Puff Count		Smoking Session Detection		
	Recall	Precision	F1	mIoU	Ground truth	Predicted	Recall	Precision	F1
P9	0.75	0.82	0.78	0.28	12	11	1.0	1.00	1.00
P10	0.94	1.00	0.97	0.57	198	194	-	-	-
P11	0.88	0.75	0.81	0.70	185	282	1.0	0.93	0.97
P12	0.84	0.83	0.84	0.65	205	204	1.0	1.00	1.00
P13	0.79	0.63	0.70	0.70	48	54	1.0	1.00	1.00
P14	0.86	0.83	0.85	0.72	172	158	1.0	1.00	1.00
P15	0.67	0.62	0.65	0.56	33	48	1.0	0.62	0.77
P16	0.75	0.75	0.75	0.44	143	178	1.0	1.00	1.00
P17	0.97	0.77	0.86	0.59	81	101	1.0	0.92	0.96
P18	0.78	0.78	0.78	0.63	146	135	1.0	1.00	1.00
P19	0.78	0.78	0.78	0.65	20	21	1.0	0.67	0.80
mean	0.82	0.78	0.80	0.59	-	-	1.00	0.92	0.95



(a)



(b)

Figure 3.9: In-wild Bland–Altman plots comparing ground truth and SmokeMon smoking topography measurements.

analyze SmokeMon energy consumption using both storage and BLE states under an opportunistic rather than continuous data collection method. In Section 3.4.1, we explained the coarse temporal segmentation we used to filter out irrelevant and highly unlikely smoking frames. We

used the same algorithm to switch the state of SmokeMon firmware from sensing only to storage or BLE. If the frame was unlikely to be smoking, we did not activate the storage or the BLE state and just used the sensing-only state. We activated the storage or the BLE state only when the algorithm detected a smoking segment. We used the in-wild data collected to simulate a realistic scenario of state change triggered by our algorithm. Then we used the state power consumption that we estimated for each state earlier, shown in Table 3.3.

Table 3.8 shows the percentage of discarded frames from each participant. On average, we achieved a reduction of 71% frames. Using the number of discarded frames, we calculated the added battery life (in hours) to SmokeMon when BLE and storage state were activated opportunistically using our algorithm. On average, when our algorithm was used to activate the BLE state opportunistically, we increased the battery life by 28% (4.1 hours), while streaming the data using BLE. Similarly, we improved the battery life by 6% (1.07 hours) on average when the storage state was activated using our algorithm. For instance, our algorithm identified 3% of the collected data as smoking related for P9, which means that we only streamed (BLE state) or stored (storage state) this portion of the data. As a result, the system was in the sensing-only state for most of the time (97%), which was the most power-efficient state.

3.6.4 SmokeMon Feedback From In-wild Participants

Below is a summary of the participants' responses to the post-experiment questionnaire noted in Section 3.5.2.2.

3.6.4.1 General Feedback.

When asked their thoughts about wearing SmokeMon to collect information about their activity, most participants (9 of 11) expressed positive comments that affirm the unobtrusive design of

Table 3.8: Further energy optimizations that can be achieved with SmokeMon. P10 is not included in this analysis because ratio of smoking events to nonsmoking events is not similar to other participants due to the participants not following the study protocol.

P	% Discarded Frames	Added Battery Life	
		BLE State	Storage State
P9	0.97	5.62	1.46
P10	-	-	-
P11	0.63	3.63	0.94
P12	0.80	4.65	1.21
P13	0.86	4.99	1.30
P14	0.48	2.78	0.72
P15	0.45	2.60	0.68
P16	0.58	3.33	0.87
P17	0.73	4.21	1.09
P18	0.69	3.99	1.04
P19	0.90	5.18	1.35
Mean	0.71	4.10	1.07

SmokeMon. Participants report that it was *"comfortable"* (P18), *"easy"* (P16), and *"not a problem"* (P11) to wear SmokeMon. Also, participants report that they *"forgot"* (P12, P10) about the device, *"didn't notice it"* (P13), and that it *"did not get in the way"* (P9). Two participants (P17, P19) report that the system was *"bulky."* The size of the system made one of those participants self-conscious and embarrassed to wear it in public (P17). When asked about improvements that can be made to SmokeMon, three participants reported that they prefer a *"smaller," "flat,"* and *"a more inconspicuous design."* Also, three participants suggested improving the magnetic attachment mechanism to make it easier to use and more stable. One participant suggested attaching the device to clothing directly rather than using magnets. In general, the experience with the initial prototype of SmokeMon was favorable. We used inexpensive off-the-shelf components to develop SmokeMon to show the method's effectiveness. In future iterations, the comments about the attachment style and the size of the system can be addressed by building a custom printed circuit board

(PCB) instead of utilizing off-the-shelf components.

3.6.4.2 *Participants Comfort.*

We asked the participants if there were any situations in which wearing the device made them feel uncomfortable. None of the participants mentioned any discomforting situation. One participant (P14) noted, "I had a moment of hesitation the first time I used the restroom while wearing the device, but then I recalled what the researcher showed me re: what the device sees and doesn't see, and after that, I wasn't concerned." We also asked participants to report situations where they took off or covered the device to further understand concerns that SmokeMon can instigate. Participants reported taking off the device while sleeping and showering, which was expected as we told them to take it off during these times. One participant (P13) reported that they would take off the device if they wanted to "lie in positions where it would get in the way." Two participants (P19, P11) mentioned that they covered the sensor with a jacket or scarf when it was too cold.

We also asked participants if SmokeMon changed how they went about their daily activities in general and specifically while smoking. Eight participants mentioned that the device did not interfere with their daily activities. The remaining three (P10, P13, P17) mentioned that they were more conscious about their everyday activities that can interfere with the device (*e.g.*, napping). Most participants (8 of 11) reported that SmokeMon did not interfere with their smoking behavior. The remaining three participants (P10, P17, P18) reported interference. However, further inspection of their comments showed that the interference was caused by their misinterpretation of the study protocol rather than the system itself. For example, two participants did not keep track of the time they wore SmokeMon, and so they wore it for more time and smoked at times that they usually did not smoke to finish the 10 hours of data collection. P10 thought that the protocol required switching on the device while smoking and turning it off otherwise, causing her to report that SmokeMon

interfered with her smoking activity as she then had to remember to switch on the device.

3.6.4.3 Bystander Reaction.

We asked participants if the device made anyone around them uncomfortable or if anyone asked them to turn off the device. Two participants mentioned that people they knew were curious about the device and what it records, which prompted participants to explain that the device captures only thermal images. None of the bystanders expressed any concerns. Instead, bystanders seemed to be worried about capturing RGB or audio data; as P12 noted, "I had one friend who inquired about what the device was recording, and he seemed satisfied when I told him it wasn't recording audio or video."

3.6.4.4 Attitude Toward Data Sharing.

After viewing their data, we asked participants how concerned they would be if their data were released to the following groups: other researchers, family, friends, doctors, dietitians, and the general public. Fig. 3.10 shows that most participants were "not at all concerned" with sharing their data with researchers, doctors, dietitians, or the general public. Two participants reported more concerns when their data were shared with friends or family but not the general public, which may suggest sensitivity of sharing smoking behavior data with people they are familiar with as opposed to just sharing thermal images about them. Overall, these results show the promise of low-resolution thermal images providing sufficient, rich data that people are willing to share with the public for the greater good.

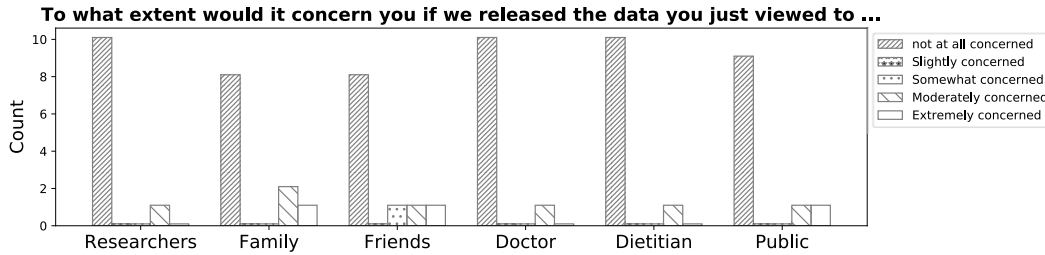


Figure 3.10: Overall, participants were not at all concerned with sharing their data.

3.7 Discussion and Future Work

This work explores the idea of using a chest-worn, low-resolution thermal camera to reliably and unobtrusively capture smoking topography. The results suggest considerable promise in using thermal cameras to detect smoking sessions and smoking topography among confounding activities while remaining energy efficient, comfortable to wear, and low burden. Below we discuss the main takeaways from our results, limitations, and future directions.

SmokeMon detects smoking sessions reliably. As shown, we are able to detect smoking sessions with high recall and precision. Similarly, smart lighters can detect smoking sessions and the frequency of smoking [109], [110]. However, if the user is using another non-smart lighter or if they are sharing the lighter with someone else, it is hard to verify if the wearer was smoking or not. In the case of SmokeMon, we are able to verify if the wearer is smoking or not, both automatically using the SmokeMon algorithm or by manually inspecting the thermal images.

SmokeMon detects smoking puffs using a single high-information-sensing modality that is comfortable to wear. Puff behaviors can vary between and within subjects, making the puff detection task very challenging. For example, as we have seen in our dataset, people may alternate which hand they use for smoking, hold the cigarette in their mouth for a long time, and engage in other non-smoking hand gestures while holding the cigarette. To overcome the puff detection challenge,

prior work has either (1) instrumented multiple low-information sensing modalities located on different parts of the body [103], [108], which may decrease adherence to wearing the system, or (2) used an uncomfortable single high-information modality sensor (*e.g.*, RGB cameras) [112]. Like RGB cameras, SmokeMon can capture challenging smoking cases without requiring additional sensor instrumentation. However, data collected by SmokeMon did not cause privacy concerns or discomfort to the wearer or bystanders.

SmokeMon unobtrusively extracts smoking topography. SmokeMon is the first system that attempts to utilize thermal sensors to extract smoking topography unobtrusively without requiring contact with the cigarette or interfering with the smoking habits. We show that, on average, SmokeMon performs as well as current smoking topography devices and ground truth. However, we believe that smoking topography results can be further improved by adding sensors to measure breathing patterns, especially when the cigarette is occluded by the user’s hand or when the user is holding the cigarette in their mouth without the hand. Researchers are developing chest sensors that can measure respiration rate and volume [130]. In the future, such sensors can be placed in the magnetic pad of SmokeMon, which has direct contact with the user’s skin.

3.7.1 Limitations

Ground truth of smoked substance: While we asked participants to label their smoking sessions, we did not ask them to identify the substance being smoked in their smoking log. This led to a confusing variability in smoking topography that affected model learning. For example, smoking topography and hand position differ between marijuana and cigarette smoking behaviors.

Generalizability: Although we show promising results, our dataset is limited, and we can not claim generalizability. An additional long-term study is needed to confirm the utility of thermal imaging

under various conditions. For example, we noticed that model performance suffered when participants smoked in a cold environment. This confusion was due to the imbalance of environmental temperatures in our dataset, as the vast majority of participants smoked in warm, indoor environments.

Chest-worn devices sensitive to wear position: Although we personalized device placement and sensor angle for each participant and added magnets for extra stability, we encountered data in which the sensor position was not ideal (*e.g.*, device tilting). Although our data augmentation method accounted for such movements, it did not handle an extreme shift in position (*i.e.*, when the device was too close to the head, resulting in an image that did not capture anything but the head). In such cases, we believe a trigger should be implemented on the device to alert the participants when the sensor is positioned incorrectly. Another approach is to increase the FoV of the camera. Unfortunately, changing the thermal camera lens is not as straightforward or as cheap as using a fisheye lens, as in the case of light-based RGB cameras. Changing the thermal image FoV requires knowledge in optics and money to manufacture the lens. However, in prior work, researchers have explored using multiple thermal cameras to increase the FoV [96]. In the case of SmokeMon, adding a second thermal camera above the current would double the vertical FoV. Adding a second camera would reduce battery life if both cameras were always on. Therefore, to reduce power consumption, the firmware should put the second camera to sleep and only trigger it to wake up during a possible smoking session (*e.g.*, when the first camera observes very high-temperature pixels).

3.7.2 Future Directions

Identifying between different smoking sessions: While our data collection focused on cigarette smoking, we noticed that many people smoke different substances and use different parapherna-

lia in addition to smoking cigarette tobacco. In some data, we observed smoking patterns from participants using electronic cigarettes (*i.e.*, "vapes") and water pipes (*i.e.*, "bongs"). Although e-cigarettes do not produce a thermal signature, the puffing behavior and the thermal reflection of the device are captured in the thermal image. Future studies in which data are collected from different smoking substances and paraphernalia would add another dimension to smoking topography beyond its current fixation on tobacco cigarettes.

Optimizing the pipeline to run on-device: As mentioned in Section 3.4.2, our puff detection pipeline benefits from using a MobileNet v2, which is capable of running on edge at a very high speed. However, deploying these models on-device often requires model compression and quantization, reducing accuracy. In the future, we aim to try our trained model on-device and, based on its real-time performance, optimize our model's architecture.

Explore different thermal sensor positions beyond the chest: In addition to the chest, we will try positioning thermal sensors on different parts of the body, including the head (*e.g.*, glasses, hat). For example, we may be able to address challenges such as occlusion by using a downward view. Other possible positions include the shoulders, where the cigarette's tip may be more visible across varying environments and body postures.

3.8 Summary of Findings

We present the development and evaluation of SmokeMon, a wearable system designed to unobtrusively measure smoking topography in free-living environments. SmokeMon uses deep learning to infer features of smoking episodes from the heat signatures captured by SmokeMon, allowing fine-grained measurement of cigarette smoking without any physical contact to the hand or cigarette. SmokeMon's battery life allows for continuous passive sensing throughout an entire waking day.

Our free-living study demonstrates the ability of SmokeMon to accurately detect smoking episodes and their topography in natural settings. Participants report high acceptability of the device and the data it collects. Overall, the results of this study show SmokeMon to have great potential in the nascent practice of measuring smoking topography.

CHAPTER 4

THERMAL-GUIDED VISUAL SAMPLING FOR EFFICIENT HAND ACTIVITY RECOGNITION

4.1 Problem Statement and Motivation

The ubiquitous nature of mobile phones and the rise of wearables is empowering researchers to redesign the study of human behavior. Recent advances in wearable cameras are expanding the utility of these cameras to enable not only visual confirmation, but also real-time detection of human activity in real-world settings [131]. The journey of wearable cameras began to scale with the popularity of visual life logging, where wearable cameras digitally capture everyday life activities through first-person images (*i.e.*, egocentric views) [132]. Wearable cameras capture rich, contextual data that can enhance machine learning models' ability to accurately classify interactions with objects and environments [3]. This capability has made them particularly valuable in activity recognition, especially for capturing hand-related behaviors [133]. Many hand-related health-risk behaviors—such as eating, smoking, drinking, and screen use—correlate with morbidity and mortality. Traditional self-report methods impose high participant burden and are prone to under- or over-reporting errors [134], [135]. In contrast, wearable cameras have enabled a more objective method to capture these hand-related activities while reducing the error and burden of recall.

Although wearable cameras hold great promise for real-world research, they face several barriers to practical implementation. Many systems record continuous video streams, resulting in redundant data and imposing heavy demands on power, storage, and computation—resources that are often scarce on embedded platforms [136], [137]. At the same time, always-on imaging raises

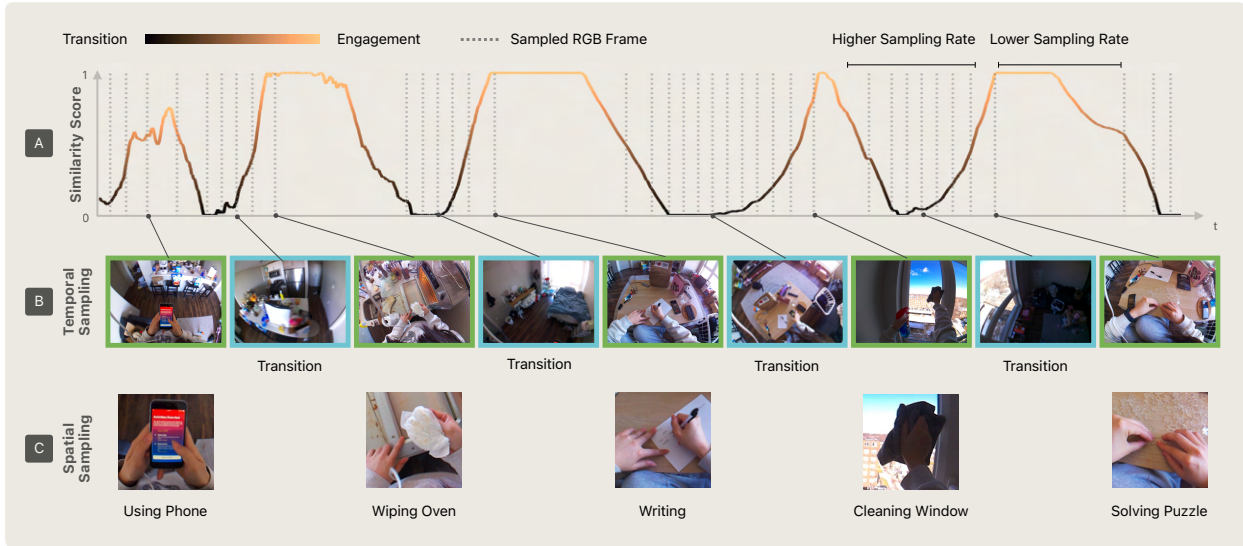


Figure 4.1: Overview of the THOR method, a spatio-temporal RGB frame sampling method that uses thermal sensing to improve the efficiency of wearable camera systems. As illustrated in (A), the method computes a similarity score as a proxy for user activity engagement. Based on this score, it increases the temporal sampling rate during activity transitions (blue bounding boxes) and decreases it during periods of continuous engagement (green bounding boxes). The temporal sampling strategy (B) selects only the most informative frames, while the spatial sampling method (C) crops the image to focus on hand-object regions, reducing the overall volume of data to be processed.

privacy concerns that can undermine user acceptance [113]. Moreover, to adhere to the Principle of Least Privilege (POLP), data collection should be restricted to only what is strictly necessary for the task at hand. To mitigate these issues, researchers have explored low-frame-rate sampling (*e.g.*, one frame every 30 seconds) [138] or event-based photo triggering using ambient light and motion sensors [138]. While these strategies improve battery life, they risk missing fine-grained hand-object interactions that are essential for accurately capturing hand-related activities (*e.g.*, eating a quick snack, or grabbing an object). Others rely on specific gesture-based triggers [131], [139] to activate the camera during automated detection of specific movements (*e.g.*, hand-to-mouth gestures) but often fail to generalize across diverse hand-related behaviors (*e.g.*, interacting with a smartphone),

and can generate false positives in real-world settings.

To mitigate the energy and storage costs of continuous recording, recent work focuses on capturing only the frames that matter. Vision-based adaptive sampling [140], [141] and video summarization techniques [142], [143] trigger RGB acquisition when salient events or scene transitions occur. Motion-activated wearable cameras and context-aware smart glasses likewise employ environmental or behavioral cues to start recording, reducing power draw while preserving key content [138], [144]. ZenCam [145] dynamically adjusts resolution and frame rate to lower bandwidth and power requirements, yet still captures the full RGB stream. Although effective, these solutions react mainly to coarse scene changes (*e.g.*, transitions from indoor to outdoor) or external triggers, and none capture the fine-grained, hand-object interactions, such as a quick snack, that are crucial for understanding health-risk behaviors.

In this work, we introduce Thermal-guided Hand-Object Reasoning (THOR), a real-time framework that leverages lower-power thermal sensing to drive adaptive RGB sampling—significantly reducing the need for continuous RGB data streams while maintaining activity recognition accuracy for both coarse- and fine-grained activity. THOR is based on two core observations: (1) *adaptive temporal sampling*: only during transitions between distinct activities is high frame-rate sampling necessary, whereas periods of sustained interaction can be captured at a much lower sampling rate; and (2) *ROI spatial sampling*: processing a cropped patch around the region of interest (ROI, around the hand and object-in-hand) can yield activity predictions on par with full-frame analysis. To our knowledge, THOR is the first egocentric vision system to employ jointly spatio-temporal adaptive sampling, reducing latency and computational demands for real-time inference, minimizing continuous RGB processing to preserve privacy, and lowering power consumption to extend device battery life (see Figure 4.1 and Figure 4.3).

We employ a lower-power, low-resolution thermal camera to continuously capture the wearer’s

hand and body poses and train a contrastive learning model on 130 distinct hand-related activities, yielding embeddings that encode pose similarities and differences. By computing the cosine similarity between the current thermal embedding and a sliding window of past embeddings, we detect activity transitions when similarity drops significantly. Upon transition detection, the RGB sensor is triggered to sample at a higher frame rate, acquiring cropped patches around the hand and the object-in-hand. Unlike prior work that merely reduces resolution to save power, our method explicitly crops the frame, focusing computational effort where it matters most. These patches are fed into a fine-tuned vision language model (VLM) to produce concise textual descriptions of the observed activity. We evaluate this spatio-temporal sampling pipeline on hand-related activity detection accuracy, real-time model inference latency, and device battery life, demonstrating substantial improvements in device efficiency and longevity.

In an in-the-wild study with 14 participants and over 30 activities and 2221 activity segments, our evaluation demonstrates that THOR reduces RGB sensor usage by 97%, decreases inference latency by 78%, and decreases system’s *overall* power consumption by 54%, all while maintaining a comparable level of accuracy (95%) to state-of-the-art vision-based methods (94%) for hand-related activity recognition. Although our prototype relies on custom hardware, the focus of this work is building and evaluating the proposed multi-modal sensing algorithm rather than hardware engineering. Embedding the system in off-the-shelf wearable devices is an obvious next step, but that integration lies beyond the scope of the present study. To summarize, the key contributions of this chapter are as follows:

- THOR I: A spatio-temporal adaptive RGB sampling method based on thermal sensing that reduces RGB sensor usage by 97% while maintaining utility (*i.e.*, high accuracy) for coarse- and fine-grained hand-related activity recognition, and extends battery life by up to 44% compared to prior work and baselines.

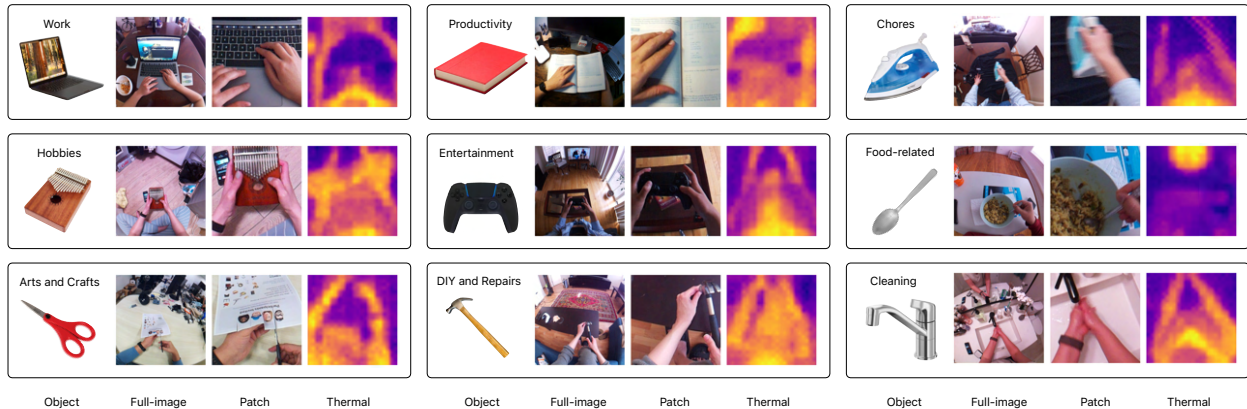


Figure 4.2: Example activity categories and associated objects from our study (15 classes shown). Our method uses a thermal camera to monitor body position during these activities and samples a cropped RGB image around the hand and object-in-hand for hand-related activity recognition. In total, we collected over 30 activity classes in an in-the-wild study.

- THOR II: A real-time end-to-end hand-related activity detection method from a single patch using a retrained (using both real and synthetic data), fine-tuned VLM (Qwen-2 VL) yielding 95% F1-score with $4.51\times$ faster inference latency compared to baseline.
- An evaluation of the proposed method in an in-wild study with 14 participants and the largest publicly available egocentric dataset, with comparisons to state-of-the-art methods.

4.2 Related Work

Recent advances in wearable sensing and egocentric vision have revolutionized hand-related activity recognition, driven by improvements in multi-modal sensor fusion, computational efficiency, and spatial understanding. The proliferation of head-mounted cameras and AR/VR devices has spurred innovations in adaptive sampling strategies to address the inherent tension between recognition accuracy and the stringent energy/compute constraints of always-on wearables. We review representative approaches across four critical dimensions: (1) adaptive frame selection, (2) multi-

modal sensing architectures, (3) egocentric hand-object interaction modeling, and (4) foundation model adaptations.

4.2.1 Adaptive Frame Selection for Efficient Action Recognition

Wearable first-person cameras operate under stringent energy and compute budgets, motivating adaptive frame-sampling strategies that process video only when new information is likely. State-of-the-art methods such as AdaFrame [140] learn to skip visually redundant frames while preserving accuracy, achieving the same top-1 recognition as dense evaluation with barely eight frames per clip—over a 60% reduction in computation [140]. Complementary ”any-time” cascades push the idea further: Ghodrati *et al.* add early-exit classifiers that classify easy clips with a handful of frames and reserve deeper inference for ambiguous cases, cutting per-video FLOPs by up to 70% without hurting performance [146]. Together, these works show that egocentric action cues are often concentrated in brief temporal windows, enabling aggressive sample-rate reduction.

A parallel line of research couples auxiliary sensors with event-driven capture to address both energy and privacy. Schiboni and Amft’s downward-angled head camera records only the wearer’s hands and torso, implicitly excluding bystanders while still capturing eating gestures [60]. Sazonov’s second-generation Automatic Ingestion Monitor (AIM-2) refines this idea: an accelerometer detects hand-to-mouth motion and triggers a still image; an on-board segmentation network then redacts faces and sensitive background regions, shrinking storage and extending battery life [147]. Beyond heuristic triggers, Possas *et al.* formulate sensor selection as a reinforcement-learning problem, switching between a low-power inertial model and a vision backbone. Their policy keeps the camera active for only 8% of runtime while matching full-camera accuracy, tripling usable battery life [148].

Unlike prior work, our system employs a low-resolution thermal sensor as a continuous sen-

tinell: thermal cues guide both *when* to sample RGB frames and *where* to crop within them. This design inherits the energy savings of sensor-triggered capture, the privacy benefits of restricting the field of view to hand-object regions, and the accuracy advantages of adaptive sampling learned in prior vision-only work. Unlike approaches driven solely by motion thresholds or fixed schedules, our system employs the thermal stream which provides real-time feedback, enabling fine-grained control over sampling rate and spatial attention precisely at moments of hand interaction.

4.2.2 Multi-Modal Sensing for Energy-Efficient Hand Activity Recognition

Wearable cameras must balance high recognition accuracy against the strict energy and compute budgets needed for extended periods of use. Adaptive frame-sampling methods tackle this challenge by evaluating only the most informative video segments. MGSampler refines this idea through motion-guided sampling that favors high-flow segments while ensuring temporal coverage, delivering both speed-ups and accuracy gains [149]. However, activities requiring static hand postures or gradual movements remain under-sampled, creating recognition gaps. Possas *et al.* model sensor scheduling as a reinforcement-learning task: their policy switches between a low-power inertial classifier and a vision backbone, sustaining accuracy with the camera active only 8% of the time [148]. However, it often optimizes short-term energy savings, leading to sub-optimal camera activation sequences during prolonged activities like meal preparation. Together, these studies show that substantial portions of egocentric video can be ignored—or processed with cheaper sensors—without harming performance.

Recent work extends this efficiency further by combining auxiliary sensor “sentinels” with adaptive visual attention. Lightweight modalities such as IMUs, audio, or low-resolution thermal imagers monitor for salient motion or sound cues and trigger high-cost vision only when needed. EgoDistill fuses sparse video with head-motion IMU data to reconstruct rich embeddings, yielding

two-to-three-fold FLOP reductions [150]. SmartAct adopts a thermal+RGB design to verify hand-to-mouth gestures while extending battery life by 30% [139]. Audio-visual SCSampler similarly deploys a fast audio network to pre-select salient clips, slashing computation $15\times$ before invoking a heavier vision model [151]. FingerTrak further demonstrates that wrist-mounted, low-resolution thermal cameras can still support accurate hand-pose estimation within tight power budgets [126]. Together, these cross-modal approaches illustrate how inexpensive cues can gate camera usage and focus computation on moments that matter. Nevertheless, because thermal signals alone cannot always disambiguate visually similar interactions, RGB imagery remains essential for broad ego-centric activity recognition despite the success of thermal-only classifiers in niche tasks such as smoking detection [152].

Once the camera is engaged, spatial and temporal attention modules prune redundancy. Hand-centric detectors that isolate the wearer’s hands and manipulated objects allow networks to crop background clutter with minimal accuracy loss [153]. Motion-guided samplers such as MGSampler continue to down-sample stagnant intervals, reserving dense inference for bursts of interaction. Privacy-aware designs fold naturally into this paradigm: ActiSight couples an RGB camera with an 8×8 thermal sensor to foreground user-specific regions while discarding bystanders [154], and HabitSense activates the RGB stream only during thermal-detected hand-to-mouth gestures, then masks the background to cut video volume by nearly 50 % while maintaining at least 90% F1-score [131]. Background masking, however, inadvertently removes contextual cues critical for disambiguating similar activities. As an example, distinguishing ”brushing teeth” from ”drinking water” requires the context of a sink versus a kitchen countertop.

We address the gap by using a low-resolution thermal stream as a continuous sentinel that guides *both* the temporal sampling of RGB frames and the spatial cropping toward hand-object regions. This unified, context-aware strategy inherits the energy savings of sensor-triggered capture, the pri-

vacy benefits of selective field-of-view restriction, and the accuracy advantages of adaptive vision attention—pushing egocentric monitoring closer to truly practical, all-day deployment.

4.2.3 Egocentric Hand-related Activity Recognition

Egocentric videos provide an unobstructed view of fine-grained hand–object interactions, yet first-person footage is rife with ego-motion, clutter, and rapid viewpoint shifts. Early work adapted third-person feature engineering to this regime: Ishihara *et al.* fuse local motion descriptors with global hand-shape cues, improving discrimination between visually similar grasps in cluttered scenes but still faltering under severe occlusion and fast camera shake[155]. Classical descriptors such as Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histograms (MBH), and dense trajectories capture rich kinematics, yet their frame-by-frame extraction remains compute-heavy and brittle when hands exit the field of view.

Deep networks replace hand-crafted descriptors with learned representations, but inherit new limitations. Multi-stream CNNs that add a dedicated hand-mask branch boost cooking-action accuracy by leveraging the frequent visibility of hands in egocentric footage [156]; however, they assume full-rate RGB and struggle when tools appear similar across tasks. Transformer-based pipelines such as IS-CLIP embed IMU motion patterns into visual tokens to reduce domain shift and enable zero-shot transfer [157], yet still require dense sampling and powerful GPUs—untenable on wearable hardware.

High-quality hand pose remains a prerequisite for fine-grained reasoning. Mueller *et al.* achieve real-time tracking from egocentric RGB-D by dividing localization and 3-D regression into separate CNN stages [158], but depth cameras are power-hungry and the method degrades with heavy object occlusion. AssemblyHands contributes three million egocentric frames with 3-D annotations, driving progress in single-view pose estimation and showing that stronger pose directly lifts

recognition performance [159]; yet current networks still mis-predict when fingers are fully hidden behind objects.

Temporal segmentation remains a key challenge. Zhang *et al.* introduce per-pixel contact masks and context-aware augmentations to improve cross-scene generalization in fine-grained hand interactions [160], while hand-hygiene pipelines first localize candidate windows using lightweight motion cues before applying two-stream CNN classifiers [161]. These methods reduce computational cost but depend on predefined boundaries or manually tuned thresholds. More recent approaches augment Slow-Fast networks with skeletal joint streams to capture spatial-temporal hand dynamics [162], yet they require high-quality upstream pose estimates and do not directly address energy-profiling constraints. While prior methods have tackled temporal segmentation and energy constraints through heuristics or multi-stage pipelines, they often depend on dense RGB data, reliable pose estimation, or hand-crafted thresholds. In contrast, our method introduces a data-efficient alternative by adaptively sampling only the most informative RGB frames based on thermal-guided transitions. By focusing on critical hand-object moments and discarding redundant footage, we significantly cut down computational cost, storage demands, and energy consumption.

4.2.4 Foundation Models for Egocentric Vision

The availability of large vision-language models such as CLIP and BLIP-2 has sparked a wave of foundation-model research tailored to first-person video. Recent efforts pre-train billions-parameter encoders on massive egocentric corpora: EgoVideo adapts a two-tower video-language architecture to 2.6 M clips from Ego4D and EPIC-Kitchens, then fine-tunes on eight EgoVis challenge tracks—requiring $16 \times$ A100 GPUs for 100 epochs [163]. Similar large-scale pre-training underlies EgoVLP and EgoNCE++, which couple contrastive video-text losses with temporal masking to learn hand-object interaction cues [164]. While these models deliver state-of-the-art retrieval

and anticipation on egocentric leaderboards, their parameter counts (0.9–2 B) and GPU footprints hamper use in consumer wearables. Even inference can exceed $40 \text{ GFLOPs} \cdot \text{s}^{-1}$, far beyond the envelope of current smart-glasses SoCs.

Foundation VLMs have also been pushed toward fine-grained hand-activity reasoning tasks. HOI-Ref extends GPT-4V to refer to the object in contact with each hand, enabling explicit question–answering about ongoing manipulations [165]. Captioning systems such as EgoInstructor retrieve instructional third-person videos to enrich first-person descriptions, improving unseen-verb coverage without manual labels [166]. VQA pipelines forecast future hand–object contacts or answer temporal queries in long clips, but nearly all rely on full-resolution RGB frames sampled at 12–30 FPS, multiplying latency and memory. When applied to 10-minute lifelog videos, a single forward pass through a billion-scale encoder can exhaust 12 GB of VRAM and consume significant energy which is untenable on embedded devices.

The gap between research prototypes and battery-constrained hardware is evident in today’s camera glasses. Meta’s Ray-Ban smart glasses last ≈ 4 hours in “moderate” photo mode but only ≈ 30 minutes with the on-device AI assistant active, as VLM inference throttles the 222 mAh cell [167], [168]. Snap’s latest Spectacles offer just 45 minutes of AR runtime before requiring a case recharge [169]. These limits highlight the need to reduce both the number of processed RGB frames and the size of the vision backbone for real-time egocentric analytics. Emerging work on retrieval-augmented captioning, cross-modal distillation, and thermal/IMU gating hints at lighter alternatives, yet a principled framework for selectively feeding only the most salient hand-centric views to foundation models remains open. **Our approach addresses this gap by pairing a thermal sentinel with aggressive RGB down-sampling, allowing hand-related activity recognition with orders-of-magnitude fewer pixels than existing egocentric vision-based pipelines.**

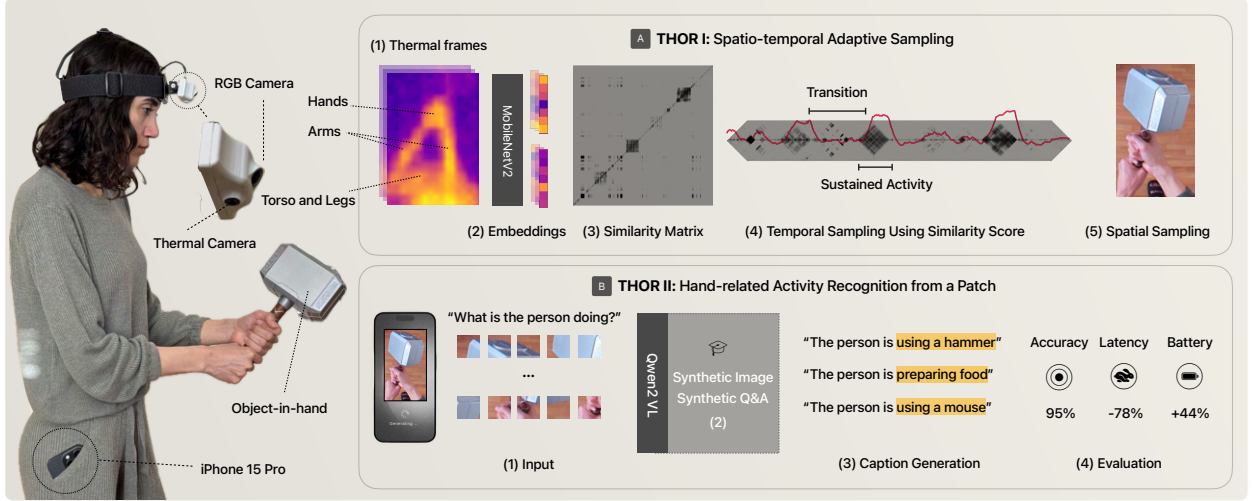


Figure 4.3: Overview of the THOR system. A wearable RGB camera is augmented with a downward-facing thermal sensor to capture hand and body positions. In THOR I (A), spatio-temporal adaptive sampling is achieved by processing thermal frames (A1) through a contrastive encoder to extract embeddings (A2), which are then compared via pairwise cosine similarity (A3) to compute a similarity score (A4) that dynamically adjusts the RGB frame rate. When RGB frames are captured, a patch centered on the hand and object-in-hand (A5) is extracted and passed to THOR II (B), which performs hand-related activity recognition. THOR II takes the patch (B1) and uses synthetic data to fine-tune a vision-language model (B2), generating free-form captions describing the user’s activity (B3). We evaluate THOR in terms of accuracy, latency, and battery lifetime.

4.3 Study Design

Most *egocentric* vision datasets rely almost exclusively on RGB data, with only a few using thermal data for niche tasks such as hand-pose estimation [170]. In contrast, we pair a low-resolution (32×24), power-efficient thermal camera with an RGB sensor to drive adaptive sampling in real time. As shown in Figure 4.3-A1, the thermal modality reliably captures heat signatures from hands, arms, and torso, across diverse activities, yielding clear cues for activity-transition detection. To build our temporal sampling backbone, we conducted an in-the-wild study with 14 participants donning a device that simultaneously records RGB and thermal data. This study was approved by

the Institutional Review Board at Northwestern University (IRB STU00223447). Each RGB frame was synchronized with a thermal frame and annotated with one of 130 hand-related activity labels. The final dataset comprises 2,221 labeled segments and is used for both training and evaluation of our model (see Figure 4.2).

4.3.1 Ego4D and Synthetic Data Generation

Foundation models typically require large-scale and diverse datasets for effective training. Although pretrained models can be fine-tuned on smaller datasets, limited variability often leads to overfitting. Moreover, collecting extensive egocentric video data is both costly and time-consuming. To address these challenges, we construct a combined training set from two sources. First, we incorporate real hand-object interaction images from Ego4D [4]. Second, we augment the training data with synthetic hand-object patches generated using a pretrained diffusion model [171]. These synthetic images depict hands interacting with everyday objects sampled from the CO3D dataset [172], which provides multi-view representations of common objects.

4.3.2 Prototype

We combine an OV5647 RGB camera and a low-resolution MLX90614 thermal sensor, both mounted on a Raspberry Pi Zero, to capture synchronized frames at 4 frames per second. The thermal module sits just below and tilted downward relative to the RGB lens for two reasons. First, to align the thermal sensor’s narrower field of view with hand- and torso-level heat signatures during head-mounted capture. Second, this orientation reduces background noise by restricting detection to the hand and object-in-hand interaction zone (see Figure 4.3).

4.3.3 Activity Selection

Our activity recognition method is activity-agnostic because it relies on a vision-language model rather than a classifier with a fixed set of classes. Despite this flexible approach, we curated 30 representative hand-centric tasks for benchmarking, drawn from prior egocentric-vision studies [173]. We balanced the list across motion types (*e.g.*, eating vs. reading) and similar gestures (*e.g.*, washing dishes vs. washing hands). To distribute coverage, each of our 14 participants was randomly assigned 15 activities, ensuring every task appeared multiple times across the cohort. This process produced a list of 30 daily activities, with varying number of segments and total duration for each activity.

4.3.4 Home-Based Protocol

To emulate real-world conditions, participants performed each assigned activity in their own homes for at least two minutes while wearing the head-mounted device. We captured uninterrupted RGB-thermal streams, using participants’ personal appliances when possible (*e.g.*, vacuums), or providing standard equipment otherwise. Transitions between activities were recorded continuously.

4.3.5 Free-Living Deployment

Beyond scripted sessions, we evaluated performance under true “in-the-wild” conditions. Participants wore the device for four hours of unstructured daily life, with brief five-minute removal breaks each hour to simulate real-world donning/doffing. This setup allowed us to assess algorithm robustness across natural activity patterns and device handling.

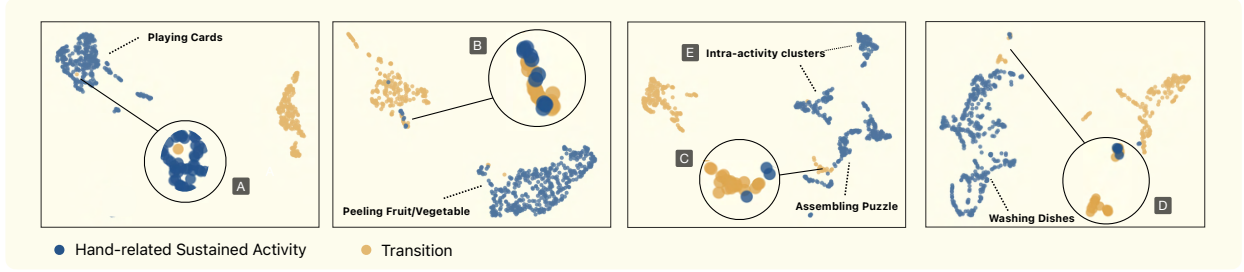


Figure 4.4: In the 2D embedding space of consecutive thermal frames produced by THOR’s temporal sampling method, hand-related activity frames (blue circles) form clusters that are separated from transition frames (orange circles). The regions A–D mark the moments where new activities begin. The smaller isolated clusters (E) represent subtle intra-activity movements (for example, minor posture shifts while assembling a puzzle).

4.4 THOR I: Adaptive Spatio-Temporal Sampling

4.4.1 Contrastive Pose Similarity Learning

To detect activity transitions, we extend contrastive learning techniques that measure image similarity between consecutive frames [174]. We adopt this idea in the thermal domain by training a model to map frames from the same activity (*i.e.*, positive pairs) close together in the embedding space, and frames from different activities (*i.e.*, negative pairs) further apart. The resulting embedding space encodes body-position variations and highlights transition points between activities (see Figure 4.4).

4.4.1.1 Model architecture

To support real-time, on-device processing, we use a lightweight feature extractor, MobileNetV2 [175], as the backbone of our adaptive sampling model. MobileNetV2 is pretrained on ImageNet [176], and we replace its classification head with a dense layer that outputs 64-dimensional (empirically determined) embeddings from each thermal frame. The cosine similarity between these embed-

dings is later used to detect transitions between activities.

4.4.1.2 Batch sampling

Effective contrastive learning hinges on careful selection of positive and negative pairs to ensure training stability [177]. We form positive pairs exclusively from frames within the same activity segment, thereby avoiding mismatches between, for example, seated versus standing portions of a single activity. Negative pairs are drawn from distinct activity classes to promote clear class separation in the embedding space. To balance training stability with training diversity, each mini-batch contains frames from 8 randomly selected activity classes, with 8 samples per class (64 frames total) sampled at each iteration.

4.4.1.3 Training and validation

We train the model using the contrastive loss function [177] on 80% of the dataset, and monitor validation accuracy on the remaining 20%. For testing, we perform leave-one-participant-out cross-validation, holding out one participant at a time to evaluate THOR I. Validation accuracy is measured using Normalized Mutual Information (NMI), which quantifies the similarity between the model’s learned clusters and the ground truth activity labels, with higher values indicating better alignment and separation of activity representations. The model achieves up to 0.64 NMI on validation set across different participants. Figure 4.4 shows the embedding space learned by the model, highlighting its ability to distinguish hand-related activity frames from transition frames. The trained model is used to extract embeddings of thermal frames in the next step.

4.4.1.4 Similarity score

We extract embeddings for each frame using the model trained in the previous step, and compute cosine similarities, using the cosine distance metric, between the current frame’s embedding and those from a fixed window of preceding frames (see Figure 4.3-A2 through A4). To mitigate noise and tailor sensitivity to the duration of different activities—ranging from short, fine-grained gestures to longer, coarse-grained episodes—we employ sliding windows of varying lengths (see Section 4.4.4.2). These comparisons yield a similarity matrix from which we derive the current similarity score (S) as the rolling-average cosine similarity over past frames. We then use S to adapt the RGB camera’s sampling rate in the next stage.

4.4.2 RGB Temporal Sampling

The new frame rate (FPS) of the RGB camera, is calculated using:

$$\text{FPS}_{\text{new}} = \text{FPS}_{\text{min}} + (\text{FPS}_{\text{max}} - \text{FPS}_{\text{min}}) \cdot F,$$

where $F = 1 - S$ is the effective similarity factor calculated using the similarity score in the previous step. FPS_{min} and FPS_{max} are two parameters that are chosen to control the number of sampled frames. To obtain S , we use a rolling window of size T over the thermal similarity matrix. For each frame index i , we compute the average similarity within a submatrix defined by the window. This rolling average is then normalized using min–max normalization across the window to produce a relative similarity score. The similarity factor S is defined as the inverse of this normalized score, such that higher activity transitions (*i.e.*, lower similarity) yield higher values of F and therefore a higher sampling rate. The complete algorithm can be found in Algorithm 2.

Algorithm 2 THOR I Dynamic FPS Update

Require: similarity matrix $\mathbf{S}$, current index i , window T , previous FPS f_{prev} , bounds $f_{\text{min}}, f_{\text{max}}$

```

1:  $W \leftarrow \{\max(0, i-T+1), \dots, i\}$ 
2: for  $j \in W$  do ▷ rolling avg. similarity
3:    $s \leftarrow \max(0, j-T+1)$ 
4:    $w_j \leftarrow \text{mean}(\mathbf{S}[s:j, s:j])$ 
5: end for
6:  $r_i \leftarrow \frac{w_i - \min_{j \in W} w_j}{\max_{j \in W} w_j - \min_{j \in W} w_j + \varepsilon}$  ▷ min-max
7:  $s \leftarrow 1 - r_i$  ▷ high change → high score
8:  $f_{\text{new}} \leftarrow f_{\text{min}} + (f_{\text{max}} - f_{\text{min}}) s$ 
9: return  $f_{\text{new}}$ 

```

4.4.3 RGB Spatial Sampling (Generating the Patch)

To limit spatial data while capturing hand-object interactions, we generate focused image patches, that represent regions of interest, guided by thermal segmentation. First, we threshold the thermal frame with Otsu’s method—separating body-heat pixels from background pixels (*e.g.*, floor)—as in [154], and produce a binary mask of heat regions. We then expand this mask to cover the relevant hand-object area. There are two main cases (either the single-hand use case, or dual-hand use case). In the single-hand use case, if the mask lies to the right of the image center, we extend it 20 pixels upward and leftward; if left of the image center, we extend 20 pixels upward and rightward. In the dual-hand use case, if the mask spans both sides of the image center, we extend 20 pixels in all horizontal directions and upward. Finally, we crop the RGB frame to this expanded region and feed the resulting patch into THOR II. This targeted spatial sampling dramatically reduces storage and compute costs while preserving essential visual cues for activity recognition.

4.4.4 Definitions

4.4.4.1 Activity Segment Bins

Our dataset includes a range of activities with varying segment lengths. Certain applications, such as activity spotting of individual feeding gestures, require the capture of short activity segments, whereas others, such as counting the number of eating episodes, involve capture of longer segments. To categorize activities based on their temporal characteristics, we adopt a data-driven approach using a head-tail classification method. This technique computes two length-based thresholds, allowing us to bin segments into three categories: short (under one minute), medium (between 1 and 2.7 minutes), and long (over 2.7 minutes) segments.

4.4.4.2 THOR Variants

We introduce three variants of our method to accommodate different segment lengths: **THOR-High**, designed for high pixel sampling to capture short segments; **THOR-Mid**, targeting medium-length segments with moderate sampling; and **THOR-Low**, which minimizes pixel usage and is optimized for capturing longer activity sequences. These variants are defined by adjusting the window parameter T and the maximum frame rate $\text{FPS}_{\max}$, as described in Algorithm 2.

4.4.5 Evaluation

4.4.5.1 Sampling Performance

We evaluate the sampling efficiency of THOR I by computing the ratio of pixels retained after our dynamic filtering to the total number of pixels that would have been captured by a fixed-rate, full-frame baseline and other baseline methods. Additionally, we assess coverage by calculating the proportion of activity segments for which THOR I captures at least four frames—ensuring sufficient

temporal context for video-based recognition models to account for potential blur in some frames.

4.4.5.2 *Baseline Methods*

We evaluate THOR I against three minimum-rate, uniform-sampling baseline methods, each tuned to guarantee at least four frames per activity segment across three duration categories (short, medium, long). For each category, we select the lowest constant sampling rate meeting this requirement: 1) Uni-Low: one frame every 17 second (to capture mainly long segments); 2) Uni-Mid: one frame every 8 seconds (to capture medium and long segments); and 3) Uni-High: one frame every 2 seconds (to capture all segments). These baselines enable a fair comparison of data-capture efficiency independent of the camera’s native frame rate.

4.4.5.3 *Power Consumption*

We measured the power consumption of key components in our system, specifically the RGB camera, thermal camera, on-device deep learning model, and the network transmission of selected patches. To capture accurate power usage, we utilized a USB power meter that directly measured the current drawn by these components during operation. These measurements were then compared to the power consumption of a baseline RGB wearable camera system, which continuously streams full-resolution images to a mobile device for processing. We also used these measurements to estimate the device’s battery life, comparing it against an RGB-only frame selection method and previous work, including an offline Python implementation of Glimpse [178]. All of the experiments were done using Raspberry Pi Zero 2 W. To ensure a fair comparison, we standardized the hardware configuration across all methods by using a single CPU core, 512 MB of RAM, and a battery with a capacity of 1500 mAh.

4.5 THOR II: Hand-related Activity Recognition from a Patch

4.5.1 Overview

The second part of our method, focuses on recognizing hand-related activities using the patches extracted from THOR I. Previous vision-based approaches for egocentric video understanding, such as TimeSformer [179] and Swin Transformers [180], often experience performance degradation when inputs are limited to a few frames or confined to small regions, such as hand-object patches. To address this limitation, we leverage recent advances in multi-modal foundation models, specifically vision-language models (VLMs), for visual recognition. Unlike conventional classification models, VLMs can generate captions for egocentric images using free-form vocabulary prompts (*i.e.*, open-set classification) enabling the recognition of a broader range of hand-related activities.

THOR II introduces techniques to enhance the utility of patches for automatic hand-related activity recognition and compares the resulting performance to prior deep learning methods. While foundation models are typically large and resource-intensive, we fine-tune a relatively lightweight VLM [181] (approximately two billion parameters) to generate concise captions focused on hand-related activities from patch inputs. This strategy yields two key benefits, illustrated in Figure 4.5: (1) using patches instead of full-frame images reduces the number of input tokens processed by the model, and (2) fine-tuning the model to produce short, targeted captions reduces the number of output tokens. Both factors contribute to significant reductions in latency and power consumption, as detailed in Section 4.6.2.5. In the following subsections, we describe the fine-tuning methodology and evaluation setup in detail.

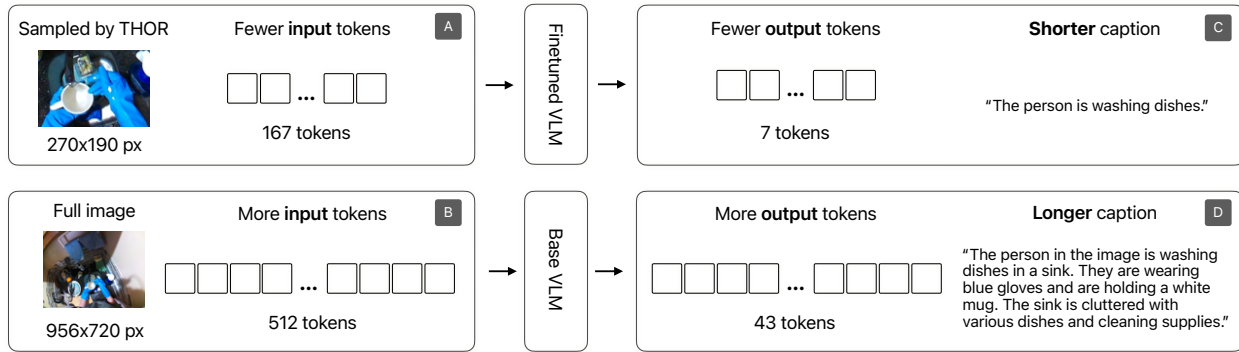


Figure 4.5: By using a low-resolution patch as input, the model processes fewer visual tokens compared to full-image input (A–B). In addition, fine-tuning the model to generate concise textual descriptions results in fewer language tokens (C–D), leading to reduced latency and lower power consumption on the device.

4.5.2 Synthetic–Augmented Teacher–Student Method

4.5.2.1 Teacher–Student Framework and Efficient Fine-tuning

For each synthetic image, we prompt a large-scale VLM (72 billion parameters) to generate three *short* question–answer pairs describing the hand-object interaction. The generated responses are post-processed and used as supervision for fine-tuning a smaller model. This teacher–student framework enables knowledge transfer from the large model to the smaller one with minimal human annotation, allowing us to use the entire dataset collected from the study as a test set. Since the fine-tuning operates on cropped patches and concise captions, both the number of input and output tokens are substantially reduced, resulting in lower computational and power demands. This design makes the approach practical for efficient real-time applications.

4.5.2.2 Model Architecture

Our objective is to develop an efficient model that maintains high recognition accuracy while remaining suitable for resource-constrained devices. Although the performance of VLMs generally

improves with increasing model size, larger models typically require powerful hardware such as high-memory GPUs. Given that our target platform is an iPhone 15 Pro, we employ a relatively lightweight model, Qwen-2 VL with 2 billion parameters [181], to generate activity captions. This choice strikes a balance between accuracy and computational efficiency, enabling practical on-device execution.

4.5.2.3 *Training details*

Training VLMs typically requires significant computational resources, including high-end GPUs. To make fine-tuning more efficient, we use Low-Rank Adaptation (LoRA) [182], which reduces the number of trainable parameters by injecting low-rank update matrices into the model. We set the rank to 8, with an alpha value of 8 and a dropout rate of 0.05. This approach speeds up training, without negatively affecting validation accuracy. To prevent overfitting, we reserve 30% of the data for validation and apply early stopping based on validation performance.

4.5.3 **Evaluation**

4.5.3.1 *Hand-related Activity Recognition*

Since the captions generated by the vision-language model are open-ended, the model is not restricted to a fixed set of predefined activities. However, for evaluation purposes, we assess performance based on 30 primary activities included in our study. One challenge is that an activity can be described many ways, for example, "using a phone" could be described as "texting," or "vacuuming" could be phrased as "cleaning the floor." To handle this variability, we manually construct a bag of keywords for each ground-truth activity. These keywords include synonymous actions or tasks commonly associated with the activity. For instance, the activity "cutting food" includes keywords such as "cutting potato" and "cutting onion," while "using a laptop" may include "working

on a laptop.” We evaluate model performance by matching generated captions to the corresponding keyword sets and report both F1-score and accuracy averaged across both participants and across activity classes.

4.5.3.2 *Baseline Methods*

We compare our method against baseline approaches that use full video data and state-of-the-art deep learning models for hand-related activity recognition. For image-based recognition, we evaluate SwinIR [180], which provides a fair comparison since our method operates on a single image. For video-based recognition, we evaluate TimeSformer [179], a leading model for spatiotemporal video understanding. Unlike our approach, which is trained on a mix of Ego4D and synthetic data, all the baseline models are trained exclusively on our dataset and are evaluated with leave-one-out cross validation.

4.5.3.3 *Generalization*

To evaluate THOR II’s ability to generalize beyond simple keyword matching, we applied our method to hand-object patches from Ego4D, the largest available egocentric dataset [4]. Because Ego4D lacks thermal imagery and only annotates hand-object interactions in a subset of videos, we first deployed a hand-object detection model [183] to extract relevant patches across the full dataset. We then ran THOR II on these patches to generate captions for over 100 annotated activity scenarios. To evaluate generalization beyond keyword-based methods, we embed the generated captions and compare their cosine similarity to the embeddings of activity narrations provided by the dataset. Higher similarity indicates more accurate captions for hand-related activities. This analysis highlights THOR II’s ability to generalize across datasets and recognize activities without relying on handcrafted keyword lists.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	Mean
THOR-H	3.42	2.79	2.32	3.45	1.87	3.87	3.42	3.37	3.67	4.39	2.44	4.45	2.15	1.31	3.07
THOR-M	1.19	0.97	0.83	1.24	0.65	1.43	1.20	1.23	1.30	1.58	0.86	1.56	0.76	0.49	1.09
THOR-L	0.29	0.23	0.20	0.29	0.16	0.33	0.29	0.29	0.31	0.38	0.20	0.37	0.18	0.12	0.26

Table 4.1: Percentage of data utilized per participant for each THOR configuration.

4.5.3.4 Latency

We measure the system’s inference latency, both with and without THOR II, in seconds to assess the impact of using patches and generating short descriptions on the model’s inference speed. Specifically, we compare the latency when using hand-object patches and fine-tuned short captions to the latency when processing full-frame images with captions generated prior to fine-tuning.

4.5.3.5 Power Consumption

We evaluated the impact of THOR on the power consumption of the user’s phone, specifically an iPhone 15 Pro with a 3274 mAh battery. Streaming full-size wearable camera images (956×720 pixels) into a VLM running on the phone in real-time could rapidly deplete the battery. However, our method reduces both the temporal and spatial amount of information processed, which in turn lowers the power required for image transfer to the phone. This reduction decreases the computational burden of VLM inference and delivers substantial power savings. We quantified these gains with an iOS app that ingests the patches output by THOR I and runs THOR II in real time directly *on-device*. We then measure the phone’s battery consumption using a USB power meter (TOL-15571 SparkFun) and compare the results against baselines, including the power consumption during inference using the model before fine-tuning.

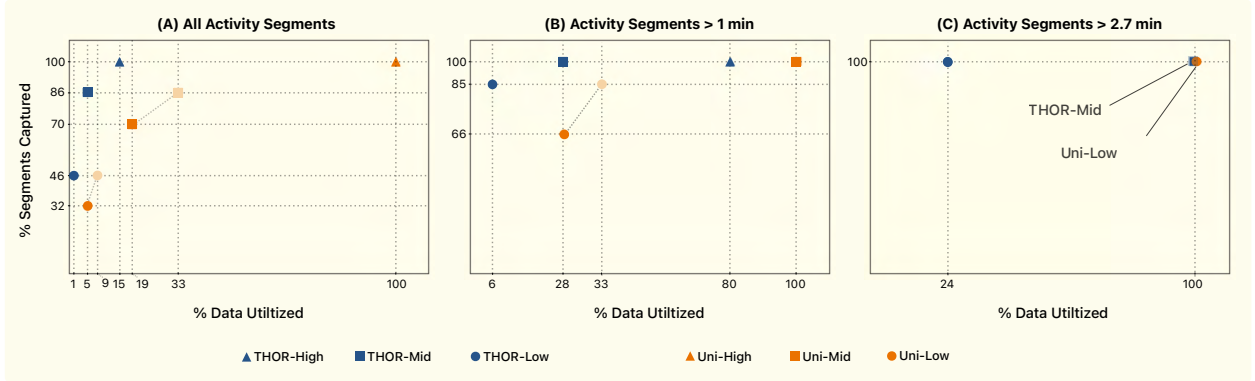


Figure 4.6: Coverage of segments with different length using different variations of THOR and baselines methods. All reported values are normalized with respect to their corresponding baselines. For instance, in (A), the percentage of segments captured by THOR is shown with Uni-High set as the 100% reference point.

4.6 Results

4.6.1 THOR I: Adaptive Saptio-temporal Sampling

4.6.1.1 Sampling Performance

Table 4.1 summarizes the proportion of data utilized by our spatio-temporal sampling approach across all participants. THOR-Low achieves complete coverage of all long activity segments using only 0.26% of the original data. By increasing data usage to 1.09% (an additional 0.83%), THOR-Mid successfully captures all medium and long segments. To achieve full segment coverage across all granularities, THOR-High requires 3.07% of the data. The variations between participants reflect differences in individual behavior patterns and activity transitions, which might be influenced by factors such as attention span. For instance, P12 exhibited frequent transitions during daily routines, resulting in the highest data usage under THOR-High (orange box in Table 4.1) to cover all segments. In contrast, Participant P14 engaged in prolonged stationary periods, often in social settings, leading to fewer short and medium segments and the lowest data consumption for capturing

short segments (green box in Table 4.1).

4.6.1.2 Comparison to Baselines

Figure 4.6 compares our spatio-temporal sampling method with uniform baseline strategies. THOR-Low captures 100% of the long activity segments while consuming only 24% of the data used by the Uni-Low baseline (Figure 4.6-C). Additionally, it captures 85% of long and medium activity segments using just 6% of the data required by Uni-Mid (Figure 4.6-B). THOR-Mid achieves complete coverage of short and medium segments while using 28% of the data consumed by Uni-Mid (Figure 4.6-B). Similarly, THOR-High captures segments of all durations, including short segments, using only 15% of the data compared to Uni-High (Figure 4.6-A). Notably, THOR-Mid consumes approximately the same amount of data as Uni-Low, yet captures 34% more medium-length segments (Figure 4.6-B). All reported values are normalized with respect to their corresponding baselines. For instance, in Figure 4.6-A, the percentage of segments captured by THOR on the y-axis is shown with Uni-High set as the 100% reference point. Overall, THOR-High achieves complete segment coverage while utilizing only 3% of the original (unfiltered) data. A detailed participant-level analysis of data usage is provided in Table 4.1.

4.6.1.3 Power Consumption

Figure 4.7-A illustrates the power consumption profile of individual system components under THOR I, compared to a baseline method that continuously streams frames from the RGB sensor. In the baseline setup, 75% of the power is consumed by the RGB sensor for frame capture, while 25% is used for transmitting frames to the phone for real-time processing. In contrast, our method's adaptive frame sampling significantly reduces the power consumption of both the RGB sensor (by 72.7%) and network transmission (by 24%). The addition of the thermal sensor and THOR I's

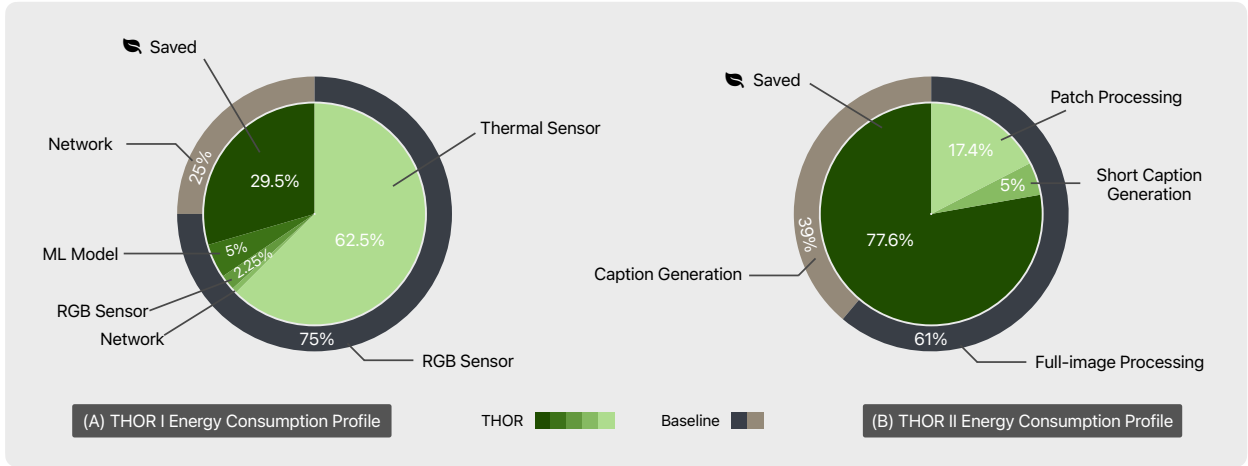


Figure 4.7: Power consumption profile of different components in THOR I (A) and THOR II (B) compared to baseline methods.

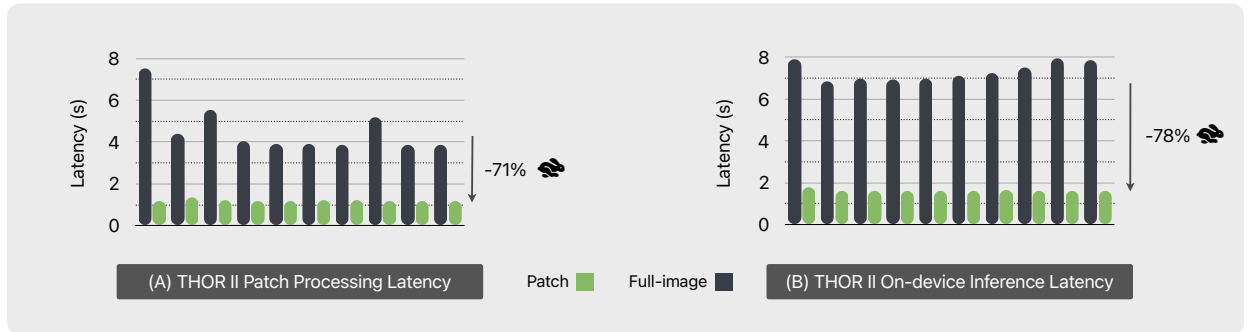


Figure 4.8: Input (A) and end-to-end (B) Latency of THOR II compared to baseline method.

deep learning model introduces minimal overhead, resulting in a net reduction of 29.5% in total system power consumption relative to the baseline.

Prior work such as Glimpse [178] proposed selecting RGB frames using a thermal sensor. An alternative approach is to trigger frame selection based on hand presence using a lightweight hand detection model [184]. Table 4.2 profiles the energy consumption of the core sensing components across THOR I, Glimpse, and RGB-only methods. Using the thermal sensor alone establishes a 300 mA baseline. Running THOR I's deep learning model increases the current draw by only 30

Component	Current (mA)	Power (W)
Thermal	300	1.50
Thermal + DL Model	330	1.77
RGB + Hand Detection	470	2.20
RGB + Network	480	2.40

Table 4.2: Power consumption of different components of the wearable device.

Setting	Current (mA)	Power (W)
THOR II Input Only	410	1.95
THOR II Input-Output	530	2.47
Baseline VLM Input Only	1450	6.82
Baseline VLM Input-Output	1520	7.11

Table 4.3: Inference energy consumption of iPhone 15 Pro under different settings.

Method	Battery Lifetime (Minute)
THOR I	270 ± 0.9
Glimpse [178]	206 ± 8.2
RGB-only	191 ± 0.2
Baseline	187 ± 0.0

Table 4.4: Device battery lifetime using different methods.

mA, indicating that the model itself is not the primary source of power drain. Adding both the RGB sensor and the hand detection model raises the current to 470 mA, and continuously streaming RGB frames to the network results in the highest draw at 480 mA. This highlights the significant cost of wireless transmission on power consumption. While Glimpse benefits from the lowest power usage by relying solely on the thermal sensor, our analysis shows that it significantly samples more RGB frames compared to our method. Therefore, as shown in Table 4.4, THOR I achieves a 31% improvement in battery life over Glimpse and 44% improvement over the baseline. Similarly, the

hand-based frame selection method requires constant hand detection processing, and as a result, the improvement in battery life over baseline is minimal, with only a 5-minute increase.

4.6.2 THOR II: Hand-related Activity Recognition from a Patch

4.6.2.1 *Hand-related Activity Recognition*

Table 4.5 reports the average performance of THOR II in recognizing 30 hand-related activities across all participants. THOR II employs THOR-High as its sampling strategy, which processes only 3% of the original video data, and achieves 96% precision, 94% recall, and 95% F1-score. These results are comparable to a full-video baseline that uses Qwen2-VL, with 94% F1-score observed for the baseline. Figure 4.10 shows the recognition accuracy of different THOR variants across activity segment durations. THOR II achieves F1-scores of 86%, 86%, and 81% under THOR-High (which detects all activities), THOR-Mid (which detects medium and long segments), and THOR-Low (which detects long segments), respectively. Each variant is evaluated on a different subset of the entire data set based on its intended sampling policy. For example, the accuracy of THOR-low (Figure 4.10) is calculated using only longer segments. In summary, when averaged across all activities, THOR-High provides performance comparable to full-video inference while using only 3% of the data, with only a 2.5% reduction in F1-score and a 2% drop in accuracy.

4.6.2.2 *Comparison to Baselines*

Table 4.5 compares the performance of THOR II against baseline methods that operate on the full video. Compared to SwinIR, which classifies activities using a single full-frame image, our method achieves a 32% higher F1-score despite operating on a single cropped patch. Relative to TimeSformer, which relies on a temporal window of frames, THOR II achieves a 15% improvement in F1-score. Qwen2-VL, which shares the same architecture as our method but uses the entire video

THOR II [†]				Qwen2-VL [‡]				SwinIR [‡]				TimeSformer [‡]			
P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc
0.96	0.94	0.95	0.94	0.96	0.93	0.94	0.94	0.73	0.55	0.63	0.60	0.81	0.79	0.80	0.79

[†] Uses patches sampled from THOR-High.

[‡] Uses entire full-resolution video input.

Table 4.5: Precision (P), Recall (R), F1-score (F1), and Accuracy (Acc) for THOR II and baseline methods.

data, achieves a comparable F1-score of 94%. These results highlight the effectiveness of multi-modal foundation models over traditional vision-based architectures. In particular, our method matches the performance of full-data methods while utilizing only 3% of the original input, enabled by the fine-tuning strategy described in Section 4.5.2.

4.6.2.3 Generalization

We evaluate the generalization ability of our method by comparing the cosine similarity between the generated captions and human-provided narrations in the Ego4D dataset. Specifically, we compute the similarity between captions produced by THOR II and narrations from two human annotators (A1 and A2). As a baseline, we also calculate the similarity between the two annotators’ narrations. As shown in Figure 4.9, the cosine similarity between A1 and A2 (0.59) is comparable to the similarity between A1 and THOR II (0.54), as well as A2 and THOR II (0.53), indicating that our method produces captions that are semantically aligned with human descriptions.

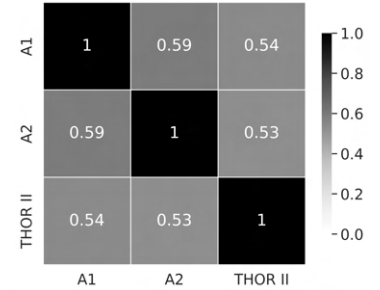


Figure 4.9: Cosine similarity between narrations provided by annotators (A1 vs A2) and our method.

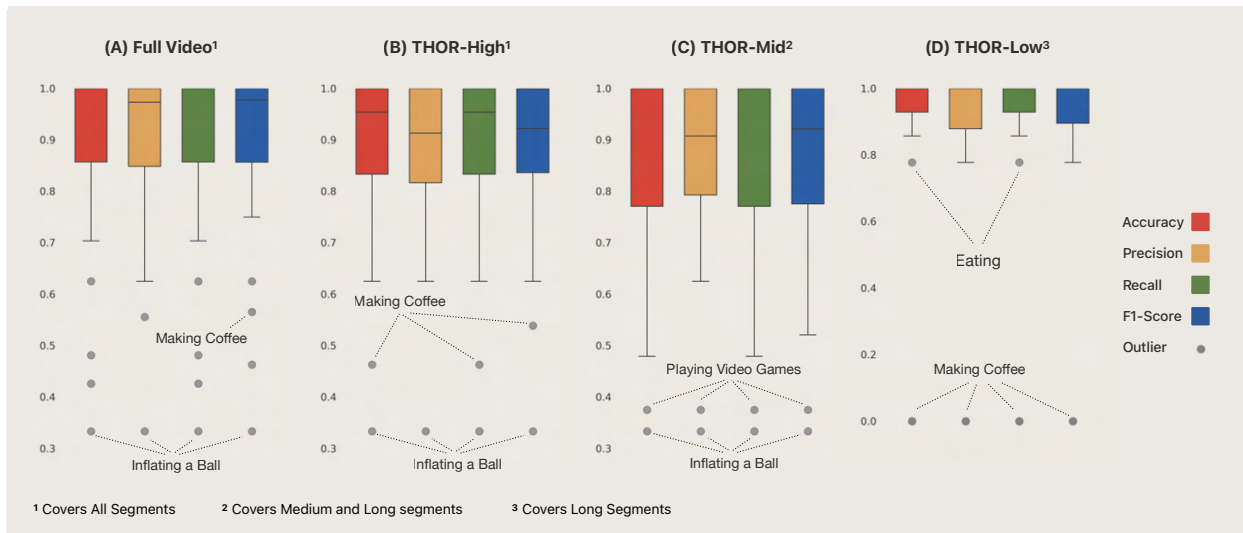


Figure 4.10: Accuracy of different THOR variants in classifying 30 hand-related activities, compared to using the full video. Note that each variant is designed for segments of different lengths; therefore, their accuracies should not be directly compared to one another. For example, the accuracy of THOR-low is calculated using only longer segments.

4.6.2.4 Latency

Figure 4.8 presents the latency profile of THOR II running on an iPhone 15 Pro. Figure 4.8-A compares the inference latency for processing full images versus hand-object patches. Using patches reduces input processing latency from 4.22 seconds to 1.21 seconds, representing a 71% decrease. This improvement is due to fewer visual tokens being processed, as shown in Figure 4.5. Figure 4.8-B shows the end-to-end latency, including caption generation. Since THOR II is fine-tuned to produce concise, hand-object-specific captions, total token generation time is shorter, reducing total latency from 6.63 seconds to 1.47 seconds per query, which corresponds to a 78% reduction.

4.6.2.5 *Power Consumption*

Figure 4.7-B shows the power consumption profile of model inference under THOR II compared to a baseline that uses the full-image input. In the baseline setup, 61% of the power is consumed during full-image processing. In contrast, our method processes only patches, reducing inference power usage to 17.4%. The shorter caption generation lowers power consumption even further by an additional 34%. Overall, our method reduces the model’s total inference power requirement by 77.6%. Table 4.3 isolates the cost of our method inference under different input conditions. Providing hand-object patches as input consumes a modest 1.95 W, while generating text on those same patches adds 21% overhead (2.47 W). In contrast, feeding the full-resolution image multiplies the power consumption by more than 3.5 $\times$ (6.82 W), and generating output on full images pushes usage to 7.11 W—over 2.9 times compared to our method. This result confirms that spatially constrained inputs cut power by up to 61% relative to full-image inference.

4.7 Discussion and Future Work

4.7.1 Potential to Integrate Video Intelligence in Existing Systems: Wearable Camera Data Browser

Despite growing interest in using wearable cameras for health-related behavior monitoring, there has been limited effort toward the development of interactive tools that facilitate efficient browsing of wearable camera footage. Currently, dietitians investigating problematic eating behaviors, or researchers conducting ground truth annotation and visual confirmation, must manually review prolonged recordings that are both redundant and computationally burdensome. These practices diminish the scalability of such data-driven studies. Our method, THOR, addresses these inefficiencies by eliminating a significant portion of video frames that are not informative for hand-related

activity recognition. This advancement opens the door to the creation of intelligent systems, such as an intelligent video browser tailored to the needs of clinical and research users. Unlike prior video browsing tools, which primarily rely on coarse visual cues such as hand presence or motion magnitude [185], our system extracts semantic descriptions from sampled patches using VLMs. These descriptions enable keyword-based search and filtering functionalities, allowing users to navigate video logs not by timestamps, but through meaningful textual summaries of the activities. By integrating THOR’s spatio-temporal sampling and captioning pipeline into a browser interface, we envision a tool where users can query specific actions (*e.g.*, ”drinking from a mug” or ”using phone while eating”) and receive direct visual access to the most relevant segments. This approach dramatically reduces the cognitive load of reviewing video data and enhances the accessibility and utility of wearable camera recordings in behavioral health, diet monitoring, and activity analysis contexts.

4.7.2 Reshaping the Future of Activity Trackers

Conventional activity tracking applications typically rely on predefined taxonomies of activities for monitoring and classification. These taxonomies, while useful for population-level studies, constrain personalization and fail to accommodate user-specific behaviors that fall outside standardized categories. Leveraging the efficient spatio-temporal sampling pipeline of THOR, we demonstrated that lightweight vision-language models (VLMs) can operate in real-time on-device, generating semantic descriptions of hand-object interactions without requiring the full RGB video stream. A key advantage of using VLMs over conventional deep learning classifiers is their ability to produce free-vocabulary, natural language captions rather than relying on a fixed set of output classes. This capability introduces a new paradigm for activity tracking in which users are no longer limited to preset activity labels. We propose a custom activity tracker application that builds on this principle.

Users can specify the activity they wish to monitor—ranging from broad behavioral categories (*e.g.*, screen time) to highly specific actions (*e.g.*, typing on a laptop while eating) and the system can detect and log instances of the specified behavior based on keyword-matching or semantic similarity with the generated captions. This approach enables a user-centered model of activity recognition that supports individualized goals and contexts, such as monitoring time spent on particular devices, habits, or multitasking behaviors.

4.7.3 Lowering the Barrier to Foundation Model Research

Recent advancements in foundation models, particularly VLMs have opened new frontiers in visual understanding, yet their computational demands present a significant barrier to entry. Fine-tuning and inference with these models often require high-end GPUs or access to distributed cloud computing infrastructure, resources that are typically available only to large academic labs or industry partners. This limitation restricts broader participation in foundation model research and hinders the democratization of innovation. In this work, by embracing the principle of least privilege we demonstrate that it is possible to use the capabilities of large-scale VLMs without overwhelming infrastructure requirements. All experiments presented in this work were conducted using four GeForce RTX 2080 GPU, a consumer-grade device released in 2018, and without reliance on commercial cloud computing services. This illustrates a compelling path forward for smaller research labs, independent investigators, and educational settings. By promoting efficiency and accessibility, our work encourages a more inclusive foundation model research ecosystem, where scientific progress is not constrained by the availability of large-scale compute.

4.7.4 Miniaturization and Glass Form Factor

Adding a thermal sensor lets THOR depend far less on its power hungry RGB camera, reducing both power consumption and processing latency—key constraints for long-term wearables, where battery life and responsiveness are critical. Our current head-mounted prototype, however, was built to maximize data fidelity, so it is bulkier than commercial smart glasses and not ideal for daily use. In the next design cycle we plan to migrate THOR into a slimmer, more ergonomic frame. Modern smart glasses, such as Meta’s Project Aria [186], already accommodate multiple imaging sensors, including RGB, eye-tracking, and mono-scene cameras. Depending on the target application, some of these sensors could be replaced by a compact thermal module without major redesign. Thermal cameras are increasingly available in compact modules that are comparable in size and shape to existing imaging sensors, making drop-in integration feasible. Embedding our dual thermal-RGB stack in a streamlined glasses-style form factor would preserve multi-modal sensing benefits while delivering a lightweight, visually unobtrusive, and socially acceptable device.

4.7.5 Assessment of Misclassified Activities

We show that our method achieves a high F1-score in detecting 30 different fine- and coarse-grained activities. However, Figure 4.10 shows that certain activities exhibit low accuracy across all variants of THOR as well as baseline methods. Below, we discuss possible reasons behind some of these failure cases and potential solutions.

4.7.5.1 *Underrepresented Objects*

We observed failure modes when participants were ”inflating a ball” using a portable air pump. The VLM often mislabels the pump as a vacuum cleaner—due to similar handle geometry—or defaults

to describing it as a generic "black object." We find that the model frequently misidentifies the air pump as a vacuum cleaner, likely due to the visual similarity in their handles, or simply describes it as a generic black object. Occasionally the model ignores the pump entirely and instead focuses on the more familiar object (*e.g.*, the ball) and generates captions such as "playing with a ball." These errors may stem from the model's lack of exposure to objects like portable air pumps during pretraining. One way to address this limitation is targeted fine-tuning of the VLM on hand-object interactions involving underrepresented tools, enabling more precise recognition and reasoning in such contexts.

4.7.5.2 *Complex Multi-step Activities*

The activity "making coffee" presents a challenge for the model due to its procedural complexity and variation in execution. This activity typically involves a series of sub-actions, such as opening a coffee pack, filling water, grinding beans, boiling water, and choosing among brewing methods—each involving distinct tools, hand movements, and visual contexts. Such activities that span multiple sub-actions makes it difficult for the model to form a consistent representation of the overall activity. Additionally, participant-specific equipment (*e.g.*, French press, pour-over, moka pot) introduces variability that fragments the model's representation. Consequently, the model may misclassify based on the most visually prominent step. Addressing this limitation may require training with more structured, multi-step activity annotations or incorporating temporal reasoning over longer sequences.

4.7.5.3 *Keyword Overlap and Semantic Ambiguity*

The activity "playing video games" often involves a gaming controller, which can serve as a distinctive visual cue. However, this activity overlaps with others that share similar contexts, such as

”using computer.” In many cases, a person using a computer may also be playing a game, and our keyword set includes terms like ”playing games” and ”using computer” for both situations. This overlap introduces ambiguity and can lead the model to misclassify one activity as the other. One possible way to address this issue is to move beyond exact keyword matching and instead compute semantic similarity between the generated captions and activity labels (see Section 4.5.3.3). This approach allows the system to recognize activities based on the overall meaning of the description, rather than relying on specific predefined terms, which can reduce confusion in overlapping scenarios.

4.7.5.4 *Concurrent Activities*

We observe several misclassifications in cases where participants perform multiple activities at the same time, or what some may characterize as secondary activities [187]. For example, in the ”eating while using a phone” scenario, the model often generates captions that describe only one of the actions, such as ”using a phone” or ”eating,” depending on which object, like a spoon or a phone, is more visually prominent in the patch. This limitation may be due to the training data, where each caption typically corresponds to a single dominant activity. As a result, THOR II can miss secondary actions in multi-tasking situations. One possible way to improve recognition in these cases is to incorporate multi-label or compositional annotations that reflect the presence of multiple hand-object interactions. Additionally, prompting strategies could be modified to encourage the model to enumerate all visible actions (*e.g.*, ”What are all the activities happening?”), rather than identifying just the most salient one. Finally, identifying the active hand, such as the one currently manipulating an object, may help determine which task the person is focused on, leading to more accurate activity descriptions.

4.7.6 Limitations and Future Work

4.7.6.1 *Activity-Boundary Segmentation*

In this work, we primarily focused on classifying activity segments and detecting their onset. In future work, we will evaluate and extend THOR to accurately delineate both start and end boundaries of each activity, enabling precise temporal segmentation of both fine- and coarse-grained activities.

4.7.6.2 *Further Optimizations*

We plan to explore lightweight VLMs with fewer parameters to enable deployment in resource-constrained settings, including both fine-tuned and zero-shot scenarios. Additionally, we will investigate thermal-only activity classification using higher-resolution thermal cameras and examine their potential to support egocentric activity recognition without relying on RGB data.

4.7.6.3 *Deployment*

Although our study with 14 participants was conducted in an uncontrolled setting, all analysis was performed offline. Several existing clinical trials and studies are incorporating wearable thermal and RGB cameras to study health-risk behaviors. Next steps include real-time THOR deployment in such large cohorts to evaluate system robustness, latency, and usability in long-term health and behavior monitoring studies.

4.8 Summary of Findings

We presented THOR, a dual-modality RGB sampling framework that adaptively modulates both when (temporal) and where (spatial) frames are captured. By leveraging thermal cues to identify transitions between hand-related activities and dynamically adjust the RGB sampling rate and spa-

tially crops to the hand-object region, yielding a 95% F1-score on hand-activity recognition. By reducing RGB data usage, THOR addresses privacy concerns, lowers power consumption, reduces inference latency, and decreases storage requirements, all while matching the performance of full-video approaches. THOR thus provides a practical and scalable solution for longitudinal wearable-camera-based studies, particularly in applications related to health and behavior monitoring.

CHAPTER 5

ENABLING GESTURE CUSTOMIZATION WITH MINIMAL TRAINING DATA

5.1 Problem Statement and Motivation

Hand gesture recognition for human-computer interaction has been studied for decades, with research and commercial applications spanning virtual and augmented reality, wearables, and smart environments [188], [189], [190], [191], [192], [193], [194], [195]. Increasingly, the research community has had a strong desire to go beyond identifying a set of predefined gestures, and to enable users to customize and define their own [196], [197], [198]. Customization brings a multitude of benefits, making gestures more memorable [199], [200], effective [201], [202], and accessible [203], [204]. However, customization is difficult because it necessitates an efficient and user-friendly data collection step (colloquially referenced as a registration or demonstration step), while also confronting the long-standing challenge of learning from limited samples, known as few-shot learning (FSL) [205]. Specifically, FSL is difficult because models must synthesize prior knowledge with a small amount of new information without overfitting [206].

Various algorithms have been explored to tackle FSL challenges in gesture recognition [207], [208], [209], [210]. For example, transfer learning [211], fine-tuning [212], and maximizing few-shots through data augmentation and generative techniques have been investigated [210]. However, the applicability of these methods is limited, especially when the originally trained gestures significantly diverge from the customized gestures. Additionally, different data types demand distinct augmentation strategies [213]. For example, augmentation methods applied to images may not be suitable for time-series sensor data [213]. Likewise, generative techniques have their own chal-

lenges, including hallucination errors [214], rendering them unreliable for data synthesis [215]. In response, meta-learning [216] has emerged as an alternative approach to address FSL challenges. Meta-learning is a training technique where a model can learn *how to learn more effectively* [217]. Although meta-learning has been applied to various classification problems [218], [219], we are the first to enable a vision-based gesture customization methodology from a single demonstration using this technique.

An equally important dimension of gesture customization is effective feature extraction, enabling users to freely create new and diverse gestures through limited demonstration. Notably, prior work in customization only supports a limited gesture set (*e.g.*, only those with intense hand movements [210]), or relies on grammar-based visual declarative scripts [220]. Often overlooked is the capability to differentiate between irrelevant hand movements (*i.e.*, background class) and intentional hand gestures. On the other hand, our approach allows users to perform gestures of various types, including one-handed, two-handed, static, and dynamic gestures, without needing to learn any complex grammar, and with the ability to exclude those hand movements that fall within the background class. We accomplish this by employing a feature extractor, notably a graph transformer model [221], which facilitates automatic feature extraction across diverse gesture types, regardless of the level of movement in the gesture (static vs. dynamic), number of hands involved (one- vs. two-handed), and camera viewpoints (egocentric [222] vs. allocentric [223]). To the best of our knowledge, we are the first to achieve this level of customization scope and flexibility, all from a single demonstration (see Section 5.4).

Initially, we pre-trained a graph transformer [221] using a publicly available dataset [224] with 20 participants. Similar to previous work [225], we use hand keypoints as an input feature to our model. Next, we employ a meta-learning algorithm coupled with meta-augmentation techniques to train a *meta-learner*. We assess the utility of our gesture customization method through a user study

and real-world applications. Initially, we gathered data from 21 participants (different from the 20 participants used in pre-training) who performed 20 distinct and varied gestures (see Figure 5.4). We leverage this dataset to assess the accuracy of our gesture customization method, achieving 94%, 94%, and 90% average accuracy for two, three, and four new gestures (respectively), again from a single demonstration.

Further, we also evaluate our method’s adaptability to different viewpoints and a background class. Specifically, we employ a technique where we perform meta-learning using gestures captured from multiple viewpoints (*i.e.*, egocentric and allocentric) and gestures captured from activities of daily living (*e.g.*, washing hands), allowing it to reject the background class and learn a view-agnostic representation of a demonstrated custom gesture. Our results show that with this method, a custom gesture demonstrated once can be accurately distinguished from random hand movements and recognized regardless of viewpoint. We further highlight the results of this evaluation in Section 5.4.

Finally, we implemented three applications: (1) a tool for designers allowing them to experiment with custom gesture recognition models (see Figure 5.6), (2) a productivity add-on that augments keyboard shortcuts with custom gestures (see Figure 5.7), and (3) a mixed reality creativity application allowing users to interact using custom gestures (see Figure 5.8). Our findings demonstrate that our gesture customization method is highly usable and learnable, and empowers users to effortlessly incorporate new gestures that are personalized to their preferences and capabilities.

To summarize, our work makes the following contributions:

- An end-to-end vision-based customization method that supports a variety of hand gestures such as static, dynamic, one-handed, and two-handed gestures, and multiple views such as egocentric and allocentric, using only one demonstration.
- A few-shot learning algorithm that combines meta-learning and meta-augmentation tech-

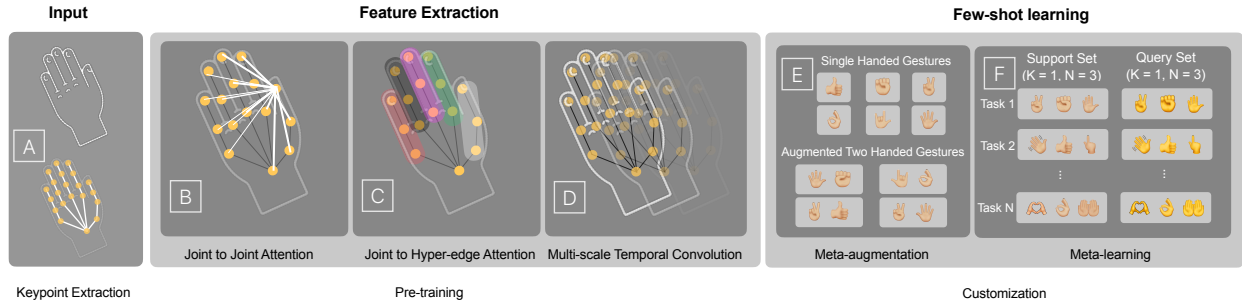


Figure 5.1: Overview of our method’s pipeline: We use a hand pose estimation model (A) to extract hand skeleton and landmarks. We utilize a graph transformer to capture both (B) joint-to-joint and (C) joint-to-group attentions. Temporal features are extracted using (D) temporal convolutions. We implement and enhance customization and few-shot learning (FSL) through (E) meta-augmentation and (F) meta-learning approaches.

niques with a graph transformer architecture.

- An evaluation of our customization method with a background class, an ablation study, and comparisons with three baseline methods.
- Implementation of three applications demonstrating the practicality of our method in a real-world setting along with key insights relevant to the interactive experience of vision-based hand gesture customization.

5.2 Related Work

5.2.1 Hand Gesture Recognition

The ability to automatically recognize hand gestures has been a long-standing research problem in the human-computer interaction community. Over the years, researchers have explored various methods to detect hand gestures, including the use of different vision-based modalities (*e.g.*, cameras, infrared ranging) [226], [227], acoustics [192], [228], and wearable inertial measurement units (IMUs) [229], [230], [231]. Red-green-blue (RGB) cameras enable the detection of a wide

variety of complex gestures and provide rich information to reliably detect hand gestures for many applications, such as sign language recognition [232].

In this work, we focus on using the RGB camera modality to provide rich information supporting a diverse range of gestures. Researchers and practitioners have extensively explored static [233] and dynamic [224] hand gesture recognition using a single RGB camera [234]. Among these works, many relied on raw RGB pixel values [235], [236], [237] as input features, and some relied on hand skeleton (*i.e.*, keypoints) [225]. Body skeleton features have been widely used for human action recognition as keypoints can be readily obtained through systems like the Kinect, LeapMotion, or robust vision-based pose estimation algorithms [238]. Skeleton-based approaches offer a more dependable alternative to traditional RGB- or depth-based methods, making these a promising choice for various real-world applications while also addressing certain computer vision challenges, like occlusion, illumination, and variations in appearance at a higher level. With the success of skeleton features in human action recognition and advancements in hand pose estimation, hand skeleton features have emerged as a promising asset for gesture recognition [225].

Many prior works in gesture recognition have been tailored to specific types of gestures, like one-handed dynamic gestures, and typically assume access to extensive datasets of predefined gestures for classifier training [239], [240]. However, there are many potential benefits of enabling user customization across a broader range of gestures, and with fewer user samples required. These advantages extend beyond increased memorability [199] and efficiency [201] to include accessibility improvements for individuals with physical disabilities [203]. Customizing gestures necessitates the model to learn from just a few user-provided samples, which poses a unique challenge, particularly for deep learning models that demand substantial data volumes. Our work tackles these challenges by leveraging advanced transformer architectures [221] and applying meta-learning algorithms [241].

5.2.2 Gesture Customization

Previous studies have developed tools that facilitate the generation of novel customized gestures (e.g., [210], [220]). The challenge with gesture customization is that it needs to support an efficient, yet effective, data collection process that creates an effortless experience for users. Although some methods require minimal training data to enable gesture recognition, they lack accuracy because they use traditional hand-crafted features [242], [243], [244], [245], [246]. Other works focus on supporting a specific type of gesture (e.g., one-handed dynamic) [210], [247], or simplify gestures by translating the static component into hand shape and the dynamic component into palm motion [220]. However, for many gestures, this simplification results in a loss of important gestural information. For instance, a *cutting scissors* gesture is dynamic, involving movement, but the palm is stationary and all movement occurs in the fingers. The *pinch drag* gesture combines both palm and finger movements simultaneously, which must both be captured for accurate recognition. Our customization system addresses this complexity, accommodating a wide array of gestures, through an end-to-end computer vision pipeline that implicitly captures both spatial and temporal features using a transformer model.

We use public datasets to avoid hand-crafting features that suffer from low accuracy for complex hand gestures. Our method further supports customization by requiring only one demonstration in the registration process. Previous work explored one-shot gesture recognition using different vision modalities [208], [248]. Unlike prior work which used fine-tuning to learn new gestures, we use meta-learning techniques to overcome the challenges of learning from a few samples.

5.2.3 Few-Shot Learning

Few-shot learning is applicable in various human-computer interaction domains, notably to enhance user experience [249], facilitate swift adaptation to user preferences [250], promote rapid

prototyping and testing [220], and effectively manage rare events or scenarios [251], [252]. Approaches to FSL can be categorized into two main groups: meta-learning-based [241], [253], [254] and non-meta-learning-based [255], [256], [257], such as transfer learning and fine-tuning. In the context of gesture customization, Xu et al. [210] adopted a non-meta-learning approach, relying on transfer learning and fine-tuning with augmented data. However, naive fine-tuning can lead to overfitting [258], and augmenting skeleton data, depending on the type of gesture, may not always yield the desired results. For instance, rotation augmentation could potentially alter the class label, transforming a *swipe right* into a *swipe up* gesture. Previous research has applied FSL to gesture recognition in various domains, including electromyography [218], vision-based [207], and WiFi-based methods [259]. In our study, we employ a model-agnostic meta-learning algorithm [241], initially designed for image classification tasks. We extend its capabilities by utilizing meta-augmentation techniques [260] to generate new gesture classes. Subsequently, we fine-tune graph transformer meta-learners, resulting in better outcomes for gesture customization.

5.3 Method

An overview of our gesture customization pipeline is shown in Figure 5.1. First, we extract hand keypoints (Section 5.3.1) and use a pre-trained graph transformer to produce hand gesture embeddings (Section 5.3.2). To adapt to custom gestures, we then further train this model in a few-shot manner using meta-learning (Section 5.3.3). This allows the model to quickly adapt when a user registers a new set of gestures. We now describe these stages in detail.

5.3.1 Keypoint Extraction

Prior research has investigated various methods for representing hands, including the use of hand skeletons. We employ a 2D hand pose estimation model [184] to extract the hand’s skeletal struc-

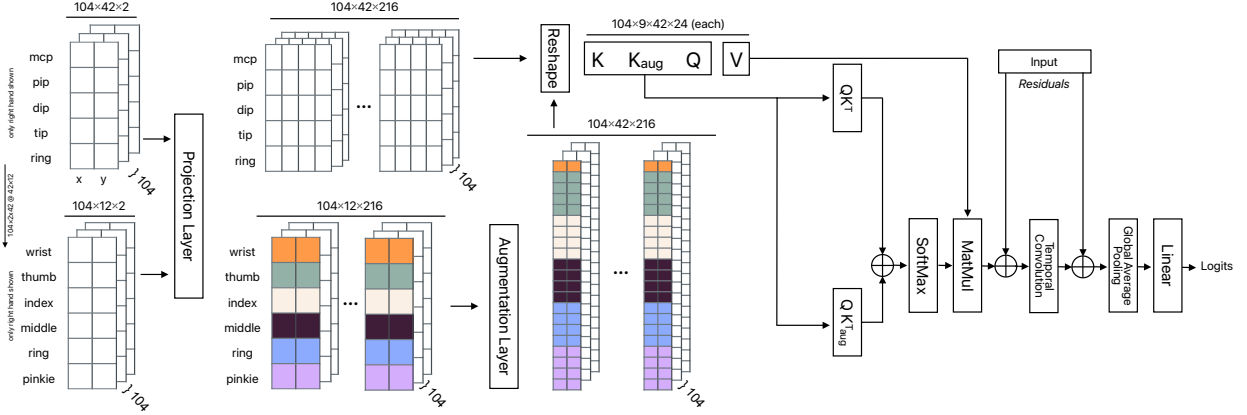


Figure 5.2: Graph transformer architecture used in our method. The graph transformer takes the keypoints and joint groups (fingers), projects them into higher dimensions and calculates attention maps to extract spatial features. The spatial and temporal features are then aggregated through temporal convolutions to yield the final output.

ture. The hand pose estimation model first detects and tracks the hand across a sequence of frames, and then extracts the keypoints for each hand detected in the frame. The skeleton comprises 21 key landmarks, including four points associated with each digit and one point for the wrist, with the location of each point specified using x - and y -coordinates. For two-handed gestures, keypoints are extracted for both hands resulting in 42 keypoints. We use these features as the input to our model.

5.3.2 Pre-Trained Feature Extractor

To lay the foundation for our gesture customization method, the initial step involves pre-training a model using a publicly accessible dataset. In this section, we describe the chosen public dataset along with the model’s architecture, and assess the performance of our pre-trained model when applied to a gesture classification task.

5.3.2.1 Dataset

The embedding space of the pre-trained model significantly influences the accuracy of the downstream task, such as customization. Consequently, it is vital to conduct pre-training using a high-quality dataset that offers diversity of classes and an abundance of samples. Additionally, collecting data from various individuals is essential to ensure that we capture variations between different subjects effectively. In recent years, several datasets have been made available to support researchers in enhancing vision-based gesture recognition techniques [233], [261], [262]. We have selected the dynamic hand gesture 14/28 dataset [263] for our work because it offers a rich assortment of 14 gesture categories characterized by diverse hand movements. Moreover, this dataset includes data from 20 individuals and provides two distinct methods for performing each gesture, such as using a single finger or the entire hand.

We intentionally exclude five gestures (specifically, *shake*, *swipe left*, *swipe right*, *rotate clockwise*, and *rotate counterclockwise*) from the training dataset because they were initially included for evaluating our customization method.

5.3.2.2 Model Architecture

As indicated in Section 5.2, prior studies have employed representations of the human body skeleton for action recognition. In our approach, we draw inspiration from this methodology, adapting a graph transformer model [221] initially designed for human action recognition. Transformers have proven to be effective in tasks that rely on temporal features, and graph transformers, in particular, excel at representing skeletons (*i.e.*, spatial information) by considering how each node (*i.e.*, joint) relates to others using an attention mechanism (Figure 5.1). We have adjusted this model to make it work effectively for recognizing hand gestures. We create two separate hand graphs, one for the left hand and one for the right hand, and then concatenate them for the transformer model. In

this graph, each node represents a specific landmark on each hand, and we establish connections between these nodes using the approach shown in Figure 5.1. All the knuckles (MCPs) are linked to the wrist point. Each knuckle in a finger is connected to the corresponding finger’s middle joint (PIP). Each middle joint (PIP) is connected to the end joint (DIP), and each end joint (DIP) is connected to the fingertip (TIP). Additionally, for every edge from some point A to some point B , an edge from B to A is also added to the graph. Moreover, the graph transformer has the ability to compute an attention score for distinct groups of nodes. These node groups are defined based on the digits of the hand, creating separate groups for the thumb, index, middle, ring, and little fingers, allowing us to generate group-level attention maps.

To accommodate a broad range of gestures, we employ a top-down approach. The transformer model processes 104 frames (approximately 1.73 seconds with 60 frames per second, empirically determined), each containing 42 keypoints. For static gestures, we perform zero-padding on the temporal dimension, and for one-handed gestures, we perform "same"-padding on the spatial dimension (*i.e.*, the keypoints of the present hand are duplicated to fill the place of the non-present hand). In contrast, for dynamic and two-handed gestures, actual data is populated in the temporal and spatial dimensions, respectively. This approach enables us to develop a single model capable of accommodating all gesture types. In the case of static gestures, which involve no motion, the system can expedite recognition by padding the input sequence with repeated frames to fulfill the 104 frames requirement. This method effectively reduces the delay for static gestures while maintaining the system’s efficiency and accuracy.

5.3.2.3 *Performance*

We use data from 15 participants for training and reserve 5 participants’ data for validation. Our graph transformer model classifies the 9 gestures with an accuracy of 86%. By employing joint-to-

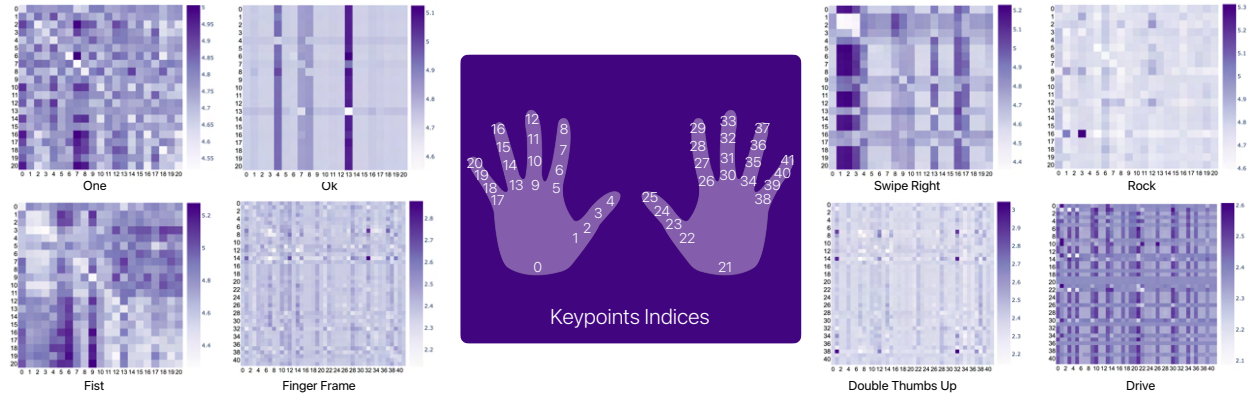


Figure 5.3: Visualization of the transformer’s joint-to-joint attention maps for eight gestures. Each map is either a 21×21 (one-handed) or 42×42 (two-handed) matrix averaged across all attention heads ($N = 9$). The model attends to different parts of the hand based on the gesture. For example, for a *swipe right* gesture, most keypoints attend to the thumb and ring finger.

joint and joint-to-group attention, the graph transformer is able to learn spatial relationships between joints and fingers. This is in contrast to graph CNNs, and was validated in our experiments, in which we found the a graph CNN to be less accurate by a 6% margin, similar to prior work [221].

Figure 5.2 highlights the process of computing these attention scores for both joint-to-joint and joint-to-group attention. In Figure 5.3 we visualize the attention matrices by averaging the attention scores across all 9 attention heads. From these visualizations we can see that the model is able to learn unique spatial characteristics for each gesture. For example, in the case of the *rock* gesture, one keypoint of the ring finger shows a high degree of attention towards one keypoint of the thumb.

5.3.3 Gesture Customization

In gesture customization, users provide a small set of their own gesture samples. Furthermore, the entire process, from registering the gesture to obtaining a trained model, should be fast, enabling users to perform multiple iterations as necessary. Deep learning models typically require substantial computation time, resources, and data. Attempting to fine-tune these models with only a handful of

samples often results in less accurate outcomes. In this section, we describe the methods we employ to address these customization challenges, including limited data availability and compatibility with various types of gestures.

5.3.3.1 Dataset

As will be detailed in Section 5.4.1, our customization dataset comprises 20 distinct gesture classes including both static, dynamic, one-handed and two-handed gestures. Static gestures involve no hand movement, whereas dynamic gestures encompass various motions of the palm, fingers, or a combination of both. Each gesture class has ten demonstrations from each of the 21 participants resulting in a total of 4200 demonstrations. In addition to these 20 gesture classes, unlike prior work, we curated another dataset of ”background” null classes, from videos of people doing everyday non-gestural activities like touching their hair, face, typing on a keyboard, washing the dishes etc. These gestures would serve as a way for our model to learn to reject the background class.

5.3.3.2 Meta-training

In order to train our model to learn from very few demonstrations, we use meta-learning, which has emerged as a popular technique for few-shot learning. We use a model-agnostic meta-learning algorithm, specifically MAML [260], to continue training our graph transformer. At a high level, meta-learning enables a model to efficiently adapt to unseen classes with very few samples.

We use the customization dataset to train meta-learners. For a few-shot gesture classification problem involving n custom gestures, the background class and k demonstrations (*i.e.*, $(n + 1)$ -way, k -shot problem) for user P , we exclude P ’s entire data from the training set along with the test split of the background class. We utilize the remaining $20 - n$ classes from other 20 participants to generate new tasks and train the meta-learner. During each training iteration, we randomly choose

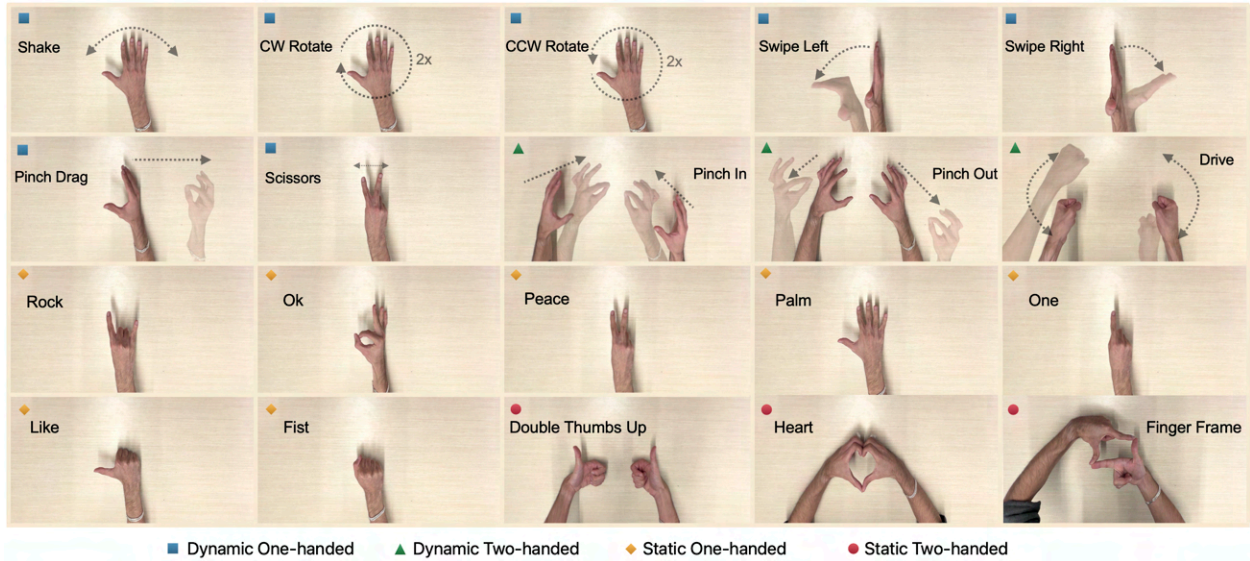


Figure 5.4: Shown here are the 20 gestures on which we evaluated our model. The blue square indicates the dynamic one-handed gestures, while the yellow diamond indicates the static one-handed gestures. Our dataset also comprises of six two-handed gestures, indicated by the green triangle for dynamic gestures and the red circle for static gestures. Our method can accommodate most gestures that a user might present from these gesture categories. This particular evaluation set was selected to represent the breadth of possible gestures, but should not be taken to suggest that our system can only accommodate these gestures.

n classes and select k samples at random from each of these n classes. We also randomly choose k samples of the background null class. These $k \times (n + 1)$ samples form the support set, which the meta-learner uses to adapt the model to the specific task. Subsequently, we draw another set of $k \times (n + 1)$ random samples from the same classes (including the background class), forming the query set (non-overlapping with support set) for testing the model's performance, and calculate the test error. The model then takes a gradient step to minimize this error. Figure 5.1F illustrates the meta-learning process. The overall effect of this is that the model learns to quickly adapt to a new set of unseen gestures.

5.3.3.3 *Meta Augmentation*

As with any deep-learning model, meta-learning is susceptible to overfitting, particularly when task diversity is limited. Augmenting is an effective method to tackle overfitting. However, classic augmentation methods, which are applied to individual data points (*e.g.*, rotation), are not applicable in meta-learning as the model is overfitting on the tasks and not a specific set of classes. This concern becomes more pronounced when dealing with two-handed gestures, where our dataset contains only six distinct classes. To mitigate this challenge, we tackle the issue by generating two-handed gesture classes. These are created by combining two separate one-handed gestures, and are treated as novel tasks for training our meta-learner. During training time, we randomly replace the padded segment of a single handed gesture with another single handed gesture, creating a novel two-handed gesture class. Figure 5.1E shows the meta-augmentation process.

5.3.3.4 *Meta-Testing*

Once the model is trained on thousands of diverse tasks involving different gestures, it is now able to classify a new task involving new custom gestures. We sample k demonstrations from each n new custom gesture classes for user P and adapt (*i.e.*, fine-tune) the model to the new gestures. Then, we test the model on the remaining samples from the same user to calculate the accuracy. It is important to note that if a user defines a custom gesture that already existed in the training set, the recognition accuracy for that specific gesture is expected to improve, while the performance of other custom gestures remains unaffected.

5.4 Evaluation

5.4.1 Data Collection

We conducted a user study to collect data from 20 new gestures to train a customized model for each individual. A versatile gesture recognition system should accommodate a diverse range of gestures, each suited for specific functions. Static one-handed gestures are effective for initiating single events, whereas dynamic two-handed gestures provide a more intuitive means for continuous variable control. For example, a pinch and drag using two hands can be used to zoom in, or manipulate object size in a virtual reality application.

Following the approach outlined in the work by Choi et al. [264], we referenced a taxonomy of static and dynamic hand gestures and selected a set of novel gestures that the pre-trained model had not encountered before. Figure 5.4 illustrates these new gestures. Our selection criteria considered factors like hand and finger movements, as well as gesture similarities. For one-handed static gestures, we included widely recognized gestures such as *thumbs up*, *palm*, *ok*, *peace*, *one*, *rock*, and *fist*, as they feature distinct finger configurations. In the realm of static two-handed gestures, we introduced *double thumbs up*, akin to the *thumbs up* gesture but performed with both hands. This choice allows us to evaluate our model's ability to distinguish between similar single and two-handed gestures. We also included *heart* and *finger frame*, two gestures involving distinct hand shapes. For dynamic gestures, we opted for gestures with similar shapes but varying hand movements, such as *swipe left* and *swipe right*, or *rotate clockwise* and *rotate counterclockwise*, as well as *shake*. Additionally, we introduced *scissors*, a gesture involving only finger movements, and *pinch drag*, *pinch in*, and *pinch out*, which feature both hand motion and hand shape changes for either one or both hands. Some gestures, like *driving* or *rotate clockwise*, lack a predefined start and end state, as users may not consistently recall these details over time. We intentionally

included these gestures to address this variability.

5.4.2 Participants and Apparatus

As part of our evaluation, we conducted a data collection study with 21 participants (3 self-identified female, 18 male, with an average age of 34 ± 9). We employed two iPhone cameras to capture gestures from multiple angles, including egocentric and allocentric. To ensure synchronization and prevent any timing discrepancies, we connected these two iPhones to a MacBook Pro and recorded the gesture videos simultaneously at 60 frames per second.

5.4.3 Design and Procedure

The data collection process consisted of four sessions. The initial two sessions focused on static gestures. In these sessions, participants sequentially performed all 10 static gestures and then repeated each gesture nine more times in a random order. This randomness removes the temporal dependency between similar gestures enabling a more rigorous evaluation. A brief break was taken midway through these sessions. After concluding the static gestures data collection, participants took another break before proceeding to the two sessions dedicated to dynamic gestures. In the case of one-handed gestures, only one hand of the participant was visible in the scene, whereas for two-handed gestures, both hands of the participant were visible. The study took 30 minutes for each participant to complete. Overall, 20 gestures were performed 10 times by 21 participants, resulting in 4200 demonstrations and approximately 500,000 frames.

5.4.4 Background Class

In order to evaluate our model’s ability to detect non-gesture background events, we also curated a dataset of background activities that users might perform day to day. We selected 600 video clips

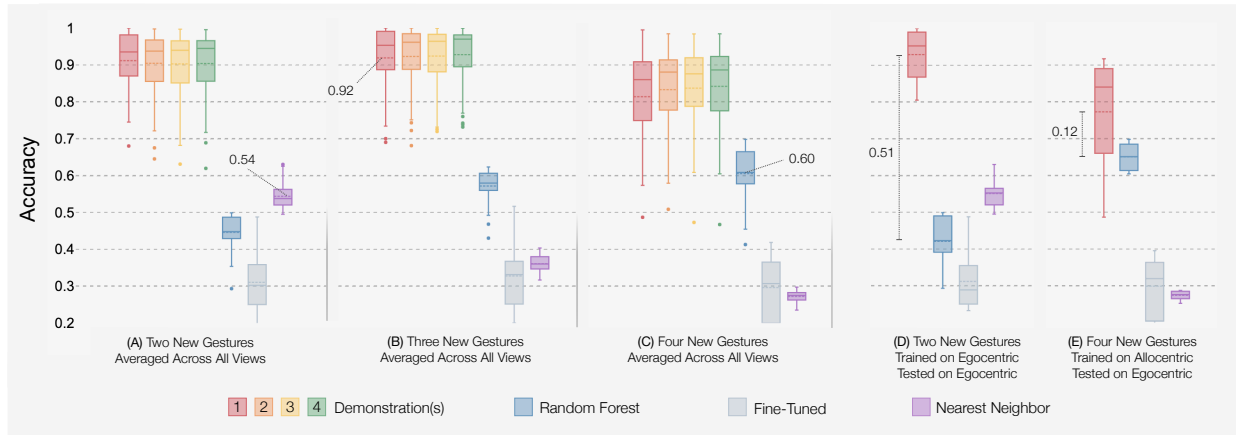


Figure 5.5: Accuracy of our method and baselines when tested with (A) two, (B) three, and (C) four new gestures along with the background class. In (D) we show a specific experiment where our method achieves its maximum accuracy improvement over baseline models such as the random forest. In (E) we show a specific experiment where our method achieves its minimum accuracy improvement over baseline models such as the random forest.

of people performing random actions like touching their hair, wiping their mouth, eating, typing on a keyboard, etc., and extracted hand keypoints for each of these clips. Similar to the method described in Section 5.3.3.3, we generated both 2 hand and single hand sequences to introduce further variance in our dataset. In total, we had 1400 sequences (150,000 frames) for training and a held out set of 360 sequences (30,000 frames) for testing. Due to the background class including hand movements that resemble gestures, gesture classification becomes particularly challenging. Nonetheless, Section 5.4.6 reveals that our method’s performance is not compromised by this additional class.

5.4.5 Training Details

To train our model, we follow the meta-learning approach outlined in Section 5.3.3. We employ a stochastic gradient descent optimizer with a fixed learning rate of 0.025, and we continue training the meta-learners until they reach convergence. As mentioned before, each experiment is characterized as an n -way k -shot problem, where n denotes the number of new gestures, and k signifies

the number of demonstrations provided in the support set. To challenge the generalizability of our method, we employ the leave-one-subject-out cross validation approach. We evaluate our approach across a range of scenarios, including $n = 2$, $n = 3$, $n = 4$, and $0 < k < 5$. To test all possible combinations of gestures would necessitate an impractical and computationally infeasible number of experiments, estimated at $\binom{20}{2} \cdot \binom{20}{3} \cdot \binom{20}{4} \cdot 21$. To address this challenge, we take a hybrid approach where we manually include certain gestures and fill in the empty spots with random gestures. In curating these experiments, we consider factors such as gesture similarity in each combination. Initially, we group together similar gestures that pose a greater challenge for differentiation. For instance, gestures like *pinch in* and *pinch out*, or *swipe left* and *swipe right*, are paired, ensuring they appear together within the customized gesture set. When customization extends to more than two classes (*e.g.*, 3 or 4), we randomly select an additional gesture(s) to complement the user’s custom gesture set. The rest of the gesture classes are used to train the meta-learner. In total we curate 30 experiments covering all the 20 gestures, and all the 21 participants in our dataset.

5.4.5.1 *Training with a Background Class*

Most prior work assume gesture candidates to be segmented before classification, leading to difference in experimental and real-world performance. Instead, we additionally train our model on classifying the n selected gestures plus a background class, which we believe more closely resembles real-world performance.

5.4.5.2 *Training with Multiple Views*

Our dataset includes gestures captured from egocentric and allocentric views. We leverage that during training, and randomly sample gestures from various viewpoints, enabling the model to learn a view-agnostic representation of the gestures. As a result, when it comes to the testing phase, the

model is able to identify user’s custom gestures from multiple views. To assess this capability, we train the model with a single demonstration from one viewpoint (*e.g.*, egocentric) and subsequently evaluate its performance using data collected from another viewpoint (*e.g.*, allocentric).

5.4.6 Model Performance

In this section, we present a summary of our customization method’s accuracy when dealing with two, three, and four new gestures and when tested with gestures captured from same view and with different views alongside the baselines. Unlike previous work, we evaluate our models with an extra background class and show that our meta-learning method outperforms baselines by a significant margin. Figure 5.5 A-C shows a condensed summary of our evaluation results.

5.4.6.1 Using Same View Point

When the model is trained and tested with a single view (*i.e.*, egocentric, allocentric) and, with just a single demonstration, our method attains 94% accuracy on average in classifying two and three new gestures. When confronted with four new gestures, our method maintains 90% accuracy. Furthermore, when we employ four demonstrations for training, the method demonstrates even higher performance, achieving average accuracy of 95% for four new gestures. An example of this improvement is plotted in Figure 5.5 A-C.

5.4.6.2 Using Different View Points

When the model is trained on a particular view and tested on a novel view, our method attains 91% accuracy on average in classifying two new gestures. Similarly, for three new gestures, our accuracy remains at 90%. When confronted with four new gestures, our method maintains 82% accuracy. Furthermore, when we employ four demonstrations for training, the method demonstrates even

higher performance, achieving average accuracy rates of 95%, 96% and 93% for two, three and four new gestures, respectively.

5.4.6.3 Baselines

We demonstrate the significance of meta-learning in facilitating gesture customization through an ablation study, in which we compare our method against three baseline approaches. First, we *fine-tune* the model with just a single demonstration of the custom gesture and subsequently evaluate its performance on test samples. Second, we employ the nearest neighbor method within the feature embedding space (size=216) of the pre-trained model. This method calculates the pair-wise distance between the embeddings of the training and test gestures and matches the closest ones to compute accuracy. Third, we employ a traditional machine learning method and measure the performance of a random forest model in classifying gestures. The model which is fine-tuned on one demonstration achieves a maximum of 64% accuracy in classifying two new gestures. Similarly, using the nearest neighbor method in the feature embedding space, the model achieves a maximum of 57% accuracy in classifying two new gestures. Thus, our customization method using meta-learning improves over these deep-learning based baselines by *at least* 30%. Similarly, our method surpasses the random forest model by an average accuracy margin of 6% across different numbers of gestures. The effectiveness of the random forest model can be attributed to its design, which is more resistant to overfitting compared to deep learning approaches. However, we show that when a non-gestural background class is introduced, adding complexity to the classification task, our method outperforms this baseline by an average of 34%.

5.4.6.4 *Evaluation with a Background Class*

As discussed in Section 5.4.4, accounting for random and non-gestural hand movements which are not included in the registered gesture set is crucial for evaluating the model’s effectiveness. Our results demonstrate that our method maintains robustness against the background class, losing a maximum of 5.5% accuracy when confronted with four new gestures. This contrast is more pronounced with other baseline methods, underscoring the superiority of our meta-learning approach. The accuracy reduction when using the baseline methods is more significant, for example, the accuracy of the random forest model is decreased 39% on average across two, three, and four gestures. This is depicted in Figure 5.5D-E.

5.4.6.5 *Registration Time vs. Accuracy*

We analyzed the balance between demonstration count and average registration time, exclusive of user interface interaction. Our findings show that the average time for registering a single gesture is 1.88 ± 0.21 seconds. Thus, we estimate that registering a gesture with two, three, and four demonstrations takes about 3.8, 5.6, and 7.5 seconds, respectively, for a mix of static and dynamic types. Our results show that for two new gestures, more demonstrations do not improve accuracy significantly, but as the number of new gestures increases, more demonstrations do improve the accuracy of the model. For example, Figure 5.5A-C illustrates the method’s accuracy with varying demonstration counts.

5.4.6.6 *Model’s latency*

We measured the training and inference speed of our model on an Apple M1 Pro CPU. The unquantized version of our method takes, on average, 20 seconds to train a model for one custom gesture

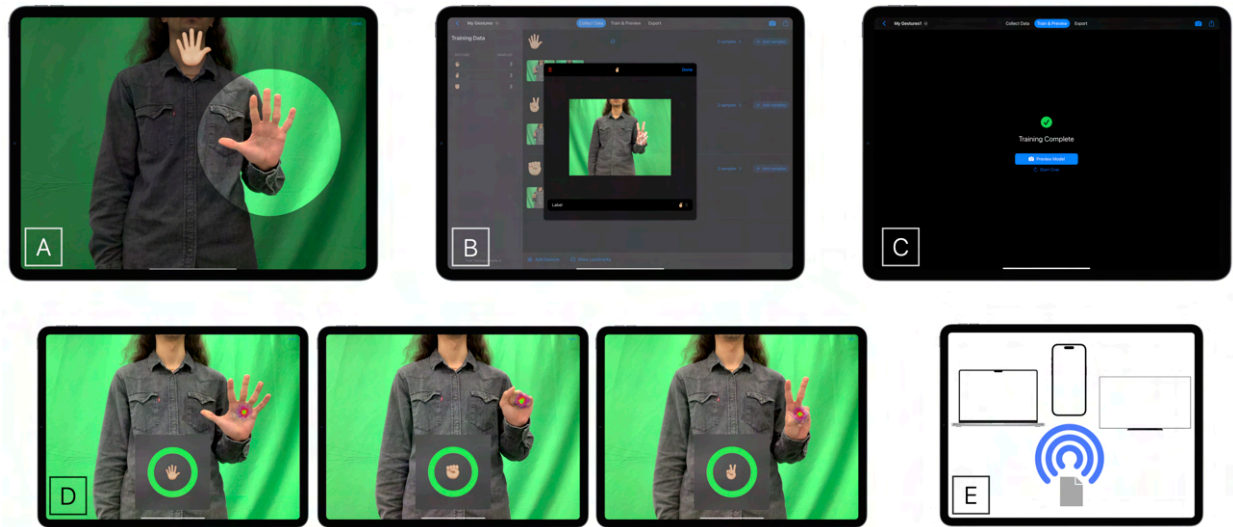


Figure 5.6: The overview of our gesture design tool. (A) The designer starts with registering their own custom gestures by demonstrating each gesture once. (B) The designer can review the samples collected, visualize the landmarks, and decide to delete, keep or add samples. (C) A graph transformer model is trained on-device using the designer’s collected samples in under a minute. (D) The designer can preview and assess the real-time performance of the model. Based on the result, the designer can decide to edit the gestures or samples by going back to step AB. (E) The custom gesture model can be further be exported to other devices or applications enabling seamless interaction and deployment.

and about one minute to train for four gestures, with each training session running for five epochs. The average inference latency is 160 milliseconds on the same machine. Our implementation uses a 1.73-second sliding window with a step size of 100 milliseconds, resulting in a total inference time of approximately 1.89 seconds per window. These parameters can be optimized to decrease training and inference times by condensing epoch lengths, adjusting window sizes, and utilizing GPUs.

5.5 Example Applications

5.5.1 Design Tool

Despite extensive research on hand gesture recognition and interaction robustness, there has been minimal emphasis on developing custom gesture design tools that require minimal data. Consequently, designers often restrict themselves to simple gestures, hindering the creation of applications, such as in game design or 3D drawing [265], [266], [267], that benefit from more complex hand gestures.

This section outlines the development of a gesture design tool aimed at facilitating the quick prototyping of hand gestures through our customization approach (see Figure 5.6). We developed an iPad application using Swift UI and other iOS frameworks such as CoreML, featuring several key functionalities.

5.5.1.1 *Registration*

This feature enables designers to define custom gestures with a single demonstration. The app achieves rapid and seamless registration by tracking hand movements and capturing motion-based samples. We designated a specific area of the screen to detect when a hand enters this zone. The app then uses the hand's motion within this area to identify the beginning and conclusion of a gesture, as illustrated in Figure 5.6A.

5.5.1.2 *Review*

Designers have the opportunity to examine and modify the gestures they have registered. During this phase, they can add, delete, or alter the gesture labels and the associated samples. This step is demonstrated in Figure 5.6B.

5.5.1.3 Training and Preview

Using the data they have collected, designers can train a model in under a minute. Then, a live preview allows designers to assess the trained model's performance in real-time (see Figure 5.6C-D).

5.5.1.4 Export and Deployment

Unlike any other design tools, our application supports exporting the trained model in a CoreML format, making it compatible with any application that accepts this format (see Figure 5.6E). This feature was used by designers to incorporate custom gestures into other applications.

We provided the tool to a group of designers at Apple Inc. who are interested in quick hand gesture prototyping for evaluation purposes. Over six months, these designers utilized the tool to develop and test custom interactive gestures. They were particularly pleased with the simplicity of creating a custom model and its integration into various applications. The possibility of enhancing the tool was also discussed. An interest in expanding the tool's capability to include body pose customization was expressed, which is feasible with minor adjustments. For instance, hand landmarks might be substituted with body landmarks to accommodate this enhancement. Currently, the tool necessitates the user's touch interaction to alternate between gestures during the registration phase. The designers recommended a more natural transition between gestures using voice commands like "next."

5.5.2 Video Editing

Creative tools, such as video editing software, often contain numerous toolbars and menus filled with a plethora of tools which can be daunting and inefficient for users to switch between. For this



Figure 5.7: (Left) An example video editing application where the user has customized a left-hand gesture to invoke the blade tool. (Right) The system also recognizes null classes and changes the tool back to the default pointer.

reason, creative professionals often memorize various non-intuitive keyboard shortcuts to quickly toggle between these tools. We envisioned a system in which the tools could be mapped to custom gestures that the user found intuitive. In this demo application, we utilize a smartphone mounted on a monitor and used Apple’s Desk View technology to track the hands on the desk. We also modified a video editor to additionally take in hand gestures as input (see Figure 5.7). The user can define an intuitive custom gesture, like the peace sign, and associate it with the action of splitting a video clip which is commonly accessible with `ctrl/cmd+k` shortcut. Similarly, the user can customize new gestures to select other tools, thereby reducing the burden of learning and memorizing a large number of awkward keyboard shortcuts. Additionally, when a null gesture (*i.e.*, fist) is recognized, the tool changes back to a default pointer.

5.5.3 Mixed Reality

Using hand gestures as a primary mode of input has become common with the prevalence of mixed reality or spatial computing devices, such as the Apple Vision Pro, Meta Quest 3, and others. While



Figure 5.8: A virtual whiteboard where users draw with a pen-shaped hand (Left) and erase with a wiping gesture (Right).

these devices can free the user from hardware such as a mouse, keyboard, or controller, that freedom can also bring a lack of precision during interactions. When using a creative application like a photo editor, many tools are available, but they are often concealed within menus and require keyboard shortcuts to efficiently access. To further complicate the issue, users may have different subsets of tools that they use most frequently. Gesture customization can be of great use in this scenario, allowing users to specify custom gestures for the tools they reach for most often. This takes advantage of hand gestures as the natural input mechanism of these mixed reality devices and avoids the difficulty of navigating complex menus.

We demonstrate the applicability of our approach to such devices by implementing a creativity app for visionOS on Apple Vision Pro (see Figure 5.8). In this demo app, the user is able to write on a virtual whiteboard simply by forming the hand into a shape as though writing with a pen. When the user wishes to erase, they can simply change their hand shape and wipe away strokes with the fingers, without the need to select the eraser from a toolbar. We believe this highlights the utility of our method in the context of these growing platforms. The versatility of our approach in recognizing gestures across diverse viewpoints also holds significant potential for this application. Our model was able to successfully use hand keypoints generated by the Apple Vision Pro, despite

the fact that this particular device and viewing angle were not present in our training data. This capability unlocks opportunities for a seamless user experience, especially in the growing domain of spatial computing, where hand gestures are a primary mode of interaction.

5.6 Discussion and Future Work

Here, we expand on the findings of our gesture customization study, discuss limitations, and prospective future directions for this work.

5.6.1 Implications for Designing Interactive Gesture Customization Experiences

Our study findings highlight the importance of incorporating specific interactive features to enhance vision-based gesture customization systems. While our guided gesture registration process simplified sample collection for users, we encountered a challenge where many users inadvertently captured samples that necessitated removal during the review process. This issue is particularly pronounced for dynamic gestures, which rely on precise temporal segmentation. Furthermore, we recognized the value of providing users with feedback on the model's performance during the gesture registration process, helping them decide whether to incorporate a new gesture. Addressing these challenges is pivotal in improving the overall efficiency and accuracy of the customization process. Notably, our approach offers the flexibility for users to augment the training data if they find the model's performance unsatisfactory. As suggested by one user, additional data collection during testing can be guided by the model's confidence in classifying a gesture. When the model lacks confidence in classifying a gesture, the system can prompt the user for confirmation. Subsequently, the model can be retrained using this supplementary data, following a methodology akin to active-learning [268]. Similar techniques have been adopted by Xu et al. [210] for gestures that are difficult to distinguish or similar to the background class. These techniques encourage users to

consider alternative gestures or provide more demonstrations to meet their accuracy needs. This can be handled in the registration process in which alternative gestures can be recommended to the user if the proposed gesture closely resembles existing custom gestures or cannot be distinguished from the background class.

5.6.2 Continuous Gesture Recognition

Determining the start and end of each gesture is crucial for continuous gesture recognition, especially when irrelevant hand movements are intermingled with custom gestures. One approach to address this challenge is to use heuristics that monitor hand acceleration and distinguish between the confidence levels of custom gestures and background class. Specifically, we employ the distance traveled by the hand's centroid on the image to trigger model inference (distance threshold determined empirically). Once the user performs a custom gesture, the confidence score for that gesture increases and decreases, providing a clear segmentation signal. Additionally, irrelevant hand movements are classified as background, enhancing the signal-to-noise ratio for segmentation. Although this chapter does not primarily focus on gesture segmentation, these techniques were incorporated into our design tool application and proven effective. Future work could explore additional kinematics-based heuristics, such as hand pose, to further improve segmentation accuracy.

5.6.3 Limitations and Future Work

5.6.3.1 Real-World Challenges

Our evaluation studies highlight how our method allows users to create their own set of new gestures. Our method performed well on both a public dataset and data from our own study, demonstrating the promise of combining hand pose estimation, graph transformers, and meta-learning for cus-

tomizing vision-based gestures. Our method can reject random hand movements (see Section 5.4.4). However, a key challenge is to differentiate one-handed from two-handed gestures when both hands are in the camera’s view. As discussed earlier, we suggest heuristic approaches that track the motion of both hands to address this issue effectively. Researchers are working on ways to address this problem using gesture segmentation and spotting techniques [269], [270], [271]. This topic area is rich with opportunities to further improve our gesture customization method in the future.

5.6.3.2 *Generalizability*

The assessment involving 21 participants substantiates the effectiveness of our approach. Nevertheless, it is crucial to acknowledge potential biases related to the study population. Notably, our study did not include participants with physical impairments, which limits our ability to evaluate the method’s performance in scenarios involving accessibility cases, such as individuals with missing fingers. Future work on gesture customization should be more inclusive to ensure that the technology meets the diverse needs of all users and accommodates a broader range of user capabilities and requirements. This inclusivity will contribute to a more comprehensive and robust evaluation of gesture-based systems and their potential applications.

5.6.3.3 *Technical Limitations*

Our method relies on off-the-shelf hand pose estimation models. This approach has advantages, as it allows issues like hand detection, hand tracking, and feature extraction to be addressed at a higher level. However, errors from these models can propagate through our pipeline and impact the performance of our gesture customization. In the future, we plan to measure and understand these errors better and evaluate pose estimation methods in conjunction with gesture recognition to enhance our system. Finally, although the chosen meta-learning algorithm played a pivotal role

in enabling gesture customization, we observed some fluctuations in accuracy across our experiments due to training instability (*e.g.*, training loss fluctuations or not reaching a steady state), as highlighted in previous research [272]. Additionally, we observed a general decline in accuracy as the number of custom gestures increased. To address these, future investigations could explore alternative meta-learning algorithms, such as metric learning, to enhance the effectiveness of gesture customization and evaluate methods on a higher number of gestures (more than four).

5.6.3.4 *Additional Applications*

We believe that our contributions in this chapter hold significant potential across various domains. Sign language recognition, for instance, relies on gesture recognition systems, and our customization method empowers users to establish a personalized and robust gesture recognition system, enhancing the accessibility and usability of such technologies.

5.7 **Summary of Findings**

In this chapter, we introduce a vision-based gesture customization method designed to empower end-users to create personalized gestures with just one demonstration. Our approach initially involves pre-training a graph transformer on a publicly available dataset featuring nine gestures. Subsequently, we present a comprehensive gesture customization method capable of accommodating static, dynamic, one-handed, and two-handed gestures that works across two different viewpoints (*i.e.*, allocentric, egocentric) and has the ability to distinguish gestures from other hand movements. Leveraging meta-learning techniques, we successfully address the challenge of learning from limited samples. In our evaluation, involving 21 participants and a set of 20 newly defined gestures, our method demonstrates an average accuracy of up to 95% with only one demonstration. We further demonstrate the usability of the real-time implementation of our method through three applications.

We believe this work can open doors for users to go beyond pre-existing gestures, empowering them to invent and incorporate new gestures tailored to their unique preferences and capabilities.

CHAPTER 6

CONCLUSION

Hand-related activities shape many aspects of daily life, influencing health behaviors, social interactions, and digital engagement. The ability to monitor these activities unobtrusively is essential for developing long-term health assessment tools and more intuitive human–computer interaction systems. Although wearable cameras and vision-based sensors offer a promising path toward continuous monitoring, persistent challenges remain. High-resolution RGB video, while informative, demands substantial energy, generates large volumes of data, and raises privacy concerns that restrict widespread, real-world adoption.

This dissertation demonstrates that these challenges can be addressed by positioning a low-power thermal sensor as the primary source of information, substantially reducing dependence on the more power-intensive RGB modality while preserving utility for fine-grained hand-activity recognition. The systems and algorithms developed here achieve reliable, real-time recognition in naturalistic settings, reaching 95% F1-score while using only a small fraction (approximately 3%) of RGB video data. Despite this dramatic reduction in visual sampling, accuracy remains comparable to approaches that rely on full-frame RGB input. These results mark a significant advancement toward privacy-preserving, energy-efficient, and always-on hand-activity recognition, helping move vision-based wearable sensing closer to the long-standing goal of ubiquitous computing.

In future, thermal-guided sensing can be extended through integration with additional modalities such as IMU signals, or acoustic sensors, enabling more robust adaptive sampling policies. The framework also opens opportunities for device miniaturization, making sensor packages com-

patible with everyday form factors such as glasses, or ear-worn devices. Advances in lightweight deep-learning architectures could further reduce computational costs, enabling continuous inference on low-power hardware. Finally, increasing the resolution and fidelity of thermal sensors may support high-accuracy thermal-only activity recognition, expanding the potential for privacy-preserving sensing systems that avoid RGB capture altogether. These opportunities highlight a broad and impactful trajectory for future work, grounded in the principle of capturing only the information needed and aligning sensing strategies with the practical constraints of real-world use.

REFERENCES

- [1] J. Ni, H. Tang, S. T. Haque, Y. Yan, and A. H. Ngu, “A survey on multimodal wearable sensor-based human action recognition,” *arXiv preprint arXiv:2404.15349*, 2024.
- [2] B. Fu, N. Damer, F. Kirchbuchner, and A. Kuijper, “Sensing technology for human activity recognition: A comprehensive survey,” *Ieee Access*, vol. 8, pp. 83 791–83 820, 2020.
- [3] A. R. Doherty et al., “Wearable cameras in health: The state of the art and future possibilities,” *American journal of preventive medicine*, vol. 44, no. 3, pp. 320–323, 2013.
- [4] K. Grauman et al., “Ego4d: Around the world in 3,000 hours of egocentric video,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 995–19 012.
- [5] E. M. Forman et al., “Ecological momentary assessment of dietary lapses across behavioral weight loss treatment: Characteristics, predictors, and relationships with weight change,” *Annals of Behavioral Medicine*, vol. 51, no. 5, pp. 741–753, 2017.
- [6] J. M. de Castro and E. Brewer, “The amount eaten in meals by humans is a power function of the number of people present,” *Physiology & Behavior*, vol. 51, no. 1, pp. 121–125, 1992.
- [7] R. Beaglehole et al., “Priority actions for the non-communicable disease crisis,” *The Lancet*, vol. 377, no. 9775, pp. 1438–1447, 2011.
- [8] H. L. Meiselman et al., “The role of context in food choice, food acceptance and food consumption,” *Frontiers in nutritional science*, vol. 3, p. 179, 2006.
- [9] H. L. Meiselman, “Obstacles to studying real people eating real meals in real situations,” 1992.
- [10] J. F. Sallis, N. Owen, and M. J. Fotheringham, “Behavioral epidemiology: A systematic framework to classify phases of research on health promotion and disease prevention,” *Annals of behavioral medicine*, vol. 22, no. 4, pp. 294–298, 2000.

- [11] N. Stroebele and J. M. De Castro, “Effect of ambience on food intake and food choice,” *Nutrition*, vol. 20, no. 9, pp. 821–838, 2004.
- [12] J. M. De Castro, “Eating behavior: Lessons from the real world of humans,” *Nutrition*, vol. 16, no. 10, pp. 800–813, 2000.
- [13] N. R. Lenard and H.-R. Berthoud, “Central and peripheral regulation of food intake and physical activity: Pathways and genes,” *Obesity*, vol. 16, no. S3, S11–S22, 2008.
- [14] H. R. Kissileff, “Where should human eating be studied and what should be measured?” *Appetite*, vol. 19, no. 1, pp. 61–68, 1992.
- [15] H. L. Meiselman, “Methodology and theory in human eating research,” *Appetite*, vol. 19, no. 1, pp. 49–55, 1992.
- [16] P. Pliner, “Let’s not throw out the barley with the dishwater: Comments on meiselman’s” methodology and theory in human eating research.”,” 1992.
- [17] E. Stellar, “Real eating and the measurement of real physiological and behavioral variables.”,” 1992.
- [18] B. J. Rolls and D. J. Shide, “Both naturalistic and laboratory-based studies contribute to the understanding of human eating behavior.”,” 1992.
- [19] H. Tuorila and L. Lähteenmäki, “When is eating “real”? response to meiselman,” *Appetite*, vol. 19, no. 1, pp. 80–83, 1992.
- [20] F. Bellisle, A. Dalix, and G. Slama, “Non food-related environmental stimuli induce increased meal intake in healthy women: Comparison of television viewing versus listening to a recorded story in laboratory settings,” *Appetite*, vol. 43, no. 2, pp. 175–180, 2004.
- [21] N. Alshurafa et al., “Counting bites with bits: Expert workshop addressing calorie and macronutrient intake monitoring,” *Journal of Medical Internet Research*, vol. 21, no. 12, e14904, Dec. 2019.
- [22] Y. C. Zhang et al., “Syncwise: Window induced shift estimation for synchronization of video and accelerometry from wearable sensors,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 3, Sep. 2020.

- [23] N. Alshurafa et al., *Counting bites with bits: Expert workshop addressing calorie and macronutrient intake monitoring*, Dec. 2019.
- [24] B. M. Bell et al., “Automatic, wearable-based, in-field eating detection approaches for public health research: a scoping review,” *npj Digital Medicine*, vol. 3, no. 1, pp. 1–14, Dec. 2020.
- [25] A. Adate et al., “Detecting screen presence with activity-oriented rgb camera in egocentric videos,” in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2022, pp. 403–408.
- [26] L. Hodges Steve Williams et al., “SenseCam: A retrospective memory aid,” in *International conference on ubiquitous computing*, 2006.
- [27] S. S. Intille, K. Larson, J. Beaudin, J. Nawyn, E. M. Tapia, and P. Kaushik, “A living laboratory for the design and evaluation of ubiquitous computing technologies,” in *Extended abstracts on Human factors in computing systems (CHI’05)*, 2005, pp. 1941–1944.
- [28] H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2847–2854.
- [29] M. Borazio and K. Van Laerhoven, “Combining wearable and environmental sensing into an unobtrusive tool for long-term sleep studies,” in *ACM SIGHIT International Health Informatics Symposium*, 2012, pp. 71–80.
- [30] Y. Nam, Y. Kim, and J. Lee, “Sleep Monitoring Based on a Tri-Axial Accelerometer and a Pressure Sensor,” *Sensors*, vol. 16, no. 5, p. 750, May 2016.
- [31] S. Bi et al., “Auracle: Detecting Eating Episodes with an Ear-mounted Sensor,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–27, Sep. 2018.
- [32] M. Mirtchouk, D. Lustig, A. Smith, I. Ching, M. Zheng, and S. Kleinberg, “Recognizing Eating from Body-Worn Sensors,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–20, Sep. 2017.
- [33] S. Zhang et al., “NeckSense: A Multi-Sensor Necklace for Detecting Eating Activities in Free-Living Conditions,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–26, Jun. 2020.

- [34] F. Harjanto, Z. Wang, S. Lu, A. C. Tsoi, and D. D. Feng, “Investigating the impact of frame rate towards robust human action recognition,” *Signal Processing*, vol. 124, pp. 220–232, 2016, Big Data Meets Multimedia Analytics.
- [35] R. Alharbi, T. Stump, N. Vafaie, A. Pfammatter, B. Spring, and N. Alshurafa, “I Can’t Be Myself: Effects of Wearable Cameras on the Capture of Authentic Behavior in the Wild,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–40, Sep. 2018.
- [36] M. Koziarski and B. Cyganek, “Impact of low resolution on image recognition with deep neural networks: An experimental study,” *International Journal of Applied Mathematics and Computer Science*, vol. 28, no. 4, 2018.
- [37] Z. Tu, Y. Ma, Z. Li, C. Li, J. Xu, and Y. Liu, *Rgbt salient object detection: A large-scale dataset and benchmark*, 2020. arXiv: 2007.03262 [cs.CV].
- [38] C. Yin, J. Chen, X. Miao, H. Jiang, and D. Chen, “Device-free human activity recognition with low-resolution infrared array sensor using long short-term memory neural network,” *Sensors*, vol. 21, no. 10, 2021.
- [39] R. Alharbi, C. Feng, S. Sen, J. Jain, J. Hester, and N. Alshurafa, “Heatsight: Wearable low-power omni thermal sensing,” in *2021 International Symposium on Wearable Computers*, ser. ISWC ’21, Virtual, USA: Association for Computing Machinery, 2021, pp. 108–112, ISBN: 9781450384629.
- [40] R. Alharbi et al., “Smokemon: Unobtrusive extraction of smoking topography using wearable energy-efficient thermal,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 4, Jan. 2023.
- [41] R. Alharbi, S. Sen, A. Ng, N. Alshurafa, and J. Hester, “Actisight: Wearer foreground extraction using a practical rgb-thermal wearable,” in *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2022, pp. 237–246.
- [42] R. Alharbi, M. Tolba, L. C. Petit, J. Hester, and N. Alshurafa, “To Mask or Not to Mask?” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–29, 2019.
- [43] S. Shahi et al., “Impacts of image obfuscation on fine-grained activity recognition in ego-centric video,” in *2022 IEEE International Conference on Pervasive Computing and Com-*

- munications Workshops and other Affiliated Events (PerCom Workshops)*, 2022, pp. 341–346.
- [44] S. Shahi, M. Pedram, G. Fernandes, and N. Alshurafa, “Smartact: Energy efficient and real-time hand-to-mouth gesture detection using wearable rgb-t,” in *2022 IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2022, pp. 1–4.
 - [45] O. Amft and G. Troster, “On-body sensing solutions for automatic dietary monitoring,” *IEEE pervasive computing*, vol. 8, no. 2, pp. 62–70, 2009.
 - [46] B. Dong and S. Biswas, “Wearable diet monitoring through breathing signal analysis,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2013, pp. 1186–1187, ISBN: 9781457702167.
 - [47] S. Sen, V. Subbaraju, A. Misra, R. Balan, and Y. Lee, “Annapurna: An automated smartwatch-based eating detection and food journaling system,” *Pervasive and Mobile Computing*, vol. 68, p. 101 259, 2020.
 - [48] E. Thomaz, I. Essa, and G. D. Abowd, “A practical approach for recognizing eating moments with wrist-mounted inertial sensing,” in *UbiComp 2015 - Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Association for Computing Machinery, Inc, Sep. 2015, pp. 1029–1040, ISBN: 9781450335744.
 - [49] T. Rahman et al., “BodyBeat: A mobile system for sensing non-speech body sounds,” in *Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services (Mobisys’14)*, Association for Computing Machinery, ISBN: 9781450327930.
 - [50] K. Yatani and K. N. Truong, “Bodyscope: A wearable acoustic sensor for activity recognition,” in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 341–350.
 - [51] A. Bedri, D. Li, R. Khurana, K. Bhuwalka, and M. Goel, “FitByte: Automatic Diet Monitoring in Unconstrained Situations Using Multimodal Sensing on Eyeglasses,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, vol. 20, New York, NY, USA: Association for Computing Machinery (ACM), Apr. 2020, pp. 1–12.
 - [52] S. Sen, V. Subbaraju, A. Misra, R. K. Balan, and Y. Lee, “The case for smartwatch-based diet monitoring,” in *IEEE International Conference on Pervasive Computing and Com-*

- munication Workshops, PerCom Workshops 2015*, Institute of Electrical and Electronics Engineers Inc., Jun. 2015, pp. 585–590, ISBN: 9781479984251.
- [53] E. Thomaz, “Practical food journaling,” in *UbiComp 2013 Adjunct - Adjunct Publication of the 2013 ACM Conference on Ubiquitous Computing*, 2013, pp. 355–360, ISBN: 9781450322157.
 - [54] K. S. Chun, S. Bhattacharya, and E. Thomaz, “Detecting eating episodes by tracking jaw-bone movements with a non-contact wearable sensor,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, Mar. 2018.
 - [55] E. Sazonov et al., “Non-invasive monitoring of chewing and swallowing for objective quantification of ingestive behavior,” *Physiological measurement*, vol. 29, no. 5, p. 525, 2008.
 - [56] S. Zhang, R. Alharbi, M. Nicholson, and N. Alshurafa, “When generalized eating detection machine learning models fail in the field,” in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, ser. UbiComp ’17, Maui, Hawaii: Association for Computing Machinery, 2017, pp. 613–622, ISBN: 9781450351904.
 - [57] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, “Image browsing, processing, and clustering for participatory sensing: Lessons from a DietSense prototype,” in *Proceedings of the 4th Workshop on Embedded Networked Sensors, EmNets 2007*, 2007, pp. 13–17, ISBN: 9781595936943.
 - [58] J. M. Echterhoff and E. J. Wang, “Par: Personal activity radius camera view for contextual sensing,” *arXiv preprint arXiv:2008.07204*, vol. -, no. -, 2020.
 - [59] W. Jia et al., “Automatic food detection in egocentric images using artificial intelligence technology,” *Public health nutrition*, vol. 22, no. 7, pp. 1168–1179, 2019.
 - [60] G. Schiboni, F. Wasner, and O. Amft, “A privacy-preserving wearable camera setup for dietary event spotting in free-living,” in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, IEEE, 2018, pp. 872–877.
 - [61] P. V. Rouast and M. T. P. Adam, “Learning deep representations for video-based intake gesture detection,” *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 6, pp. 1727–1737, Jun. 2020.

- [62] S. Alletto, G. Serra, S. Calderara, F. Solera, and R. Cucchiara, “From ego to nos-vision: Detecting social relationships in first-person views,” in *Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, 2014, pp. 580–585.
- [63] A. D. Shetty, B. Shubha, K. Suryanarayana, et al., “Detection and tracking of a human using the infrared thermopile array sensor—“grid-eye”,” in *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, IEEE, 2017, pp. 1490–1495.
- [64] C. Cattuto, W. V. den Broeck, A. Barrat, V. Colizza, J.-F. Pinton, and A. Vespignani, “Dynamics of person-to-person interactions from distributed RFID sensor networks,” *PLoS ONE*, vol. 5, no. 7, C. Neylon, Ed., e11596, Jul. 2010.
- [65] M. Atzmueller, L. Thiele, G. Stumme, and S. Kauffeld, “Analyzing group interaction on networks of face-to-face proximity using wearable sensors,” in *International Conference on Future IoT Technologies (Future IoT)*, IEEE, 2018, pp. 1–10.
- [66] P. Barsocchi, A. Crivello, M. Girolami, and F. Mavilia, “Detecting social interactions in indoor environments with the red-HuP algorithm,” in *International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, IEEE, Mar. 2020.
- [67] B. M. Heravi, J. L. Gibson, S. Hailes, and D. Skuse, “Playground social interaction analysis using bespoke wearable sensors for tracking and motion capture,” in *Proceedings of the 5th International Conference on Movement and Computing*, 2018, pp. 1–8.
- [68] F. Adib and D. Katabi, “See through walls with wifi!” In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, 2013, pp. 75–86.
- [69] A. Fathi, J. K. Hodgins, and J. M. Rehg, “Social interactions: A first-person perspective,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 1226–1233.
- [70] J. Zhao, R. Han, Y. Gan, L. Wan, W. Feng, and S. Wang, “Human identification and interaction detection in cross-view multi-person videos with wearable cameras,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2608–2616.
- [71] A. Okuno and Y. Sumi, “Social activity measurement by counting faces captured in first-person view lifelogging video,” in *Proceedings of the 10th Augmented Human International Conference 2019*, 2019, pp. 1–9.

- [72] D. Haines, *How long does a gopro battery last*, <https://clicklikethis.com/how-long-does-a-gopro-battery-last/>, [Online; accessed 8-April-2021], Sep. 2020.
- [73] R. Likamwa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl, “Energy characterization and optimization of image sensing toward continuous mobile vision,” in *MobiSys 2013 - Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 2013, pp. 69–81, ISBN: 9781450316729.
- [74] D. J. Bernstein, “The salsa20 family of stream ciphers,” in *New stream cipher designs*, Springer, 2008, pp. 84–97.
- [75] S. Yang, P. Luo, C. C. Loy, and X. Tang, “Wider face: A face detection benchmark,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [76] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” *arXiv preprint arXiv:1406.2199*, 2014.
- [77] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, *Retinaface: Single-stage dense face localisation in the wild*, 2019. arXiv: 1905.00641 [cs.CV].
- [78] S. Sen, V. Subbaraju, A. Misra, R. Balan, and Y. Lee, “Annapurna: An automated smartwatch-based eating detection and food journaling system,” *Pervasive and Mobile Computing*, vol. 68, p. 101 259, 2020.
- [79] S. Zhang et al., “Necksense: A multi-sensor necklace for detecting eating activities in free-living conditions,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–26, 2020.
- [80] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, “Finding a ”kneedle” in a haystack: Detecting knee points in system behavior,” in *2011 31st International Conference on Distributed Computing Systems Workshops*, 2011, pp. 166–171.
- [81] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [82] E. Thomaz, A. Parnami, I. Essa, and G. D. Abowd, “Feasibility of identifying eating moments from first-person images leveraging human computation,” in *Proceedings of the 4th International SenseCam & Pervasive Imaging Conference on - SenseCam ’13*, ACM Press, 2013.

- [83] L. Arab, D. Estrin, D. H. Kim, J. Burke, and J. Goldman, “Feasibility testing of an automated image-capture method to aid dietary recall,” *European Journal of Clinical Nutrition*, vol. 65, no. 10, pp. 1156–1162, May 2011.
- [84] R. Alharbi, M. Tolba, L. C. Petit, J. Hester, and N. Alshurafa, “To mask or not to mask? balancing privacy with visual confirmation utility in activity-oriented wearable cameras,” *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 3, no. 3, pp. 1–29, 2019.
- [85] *Smoking & tobacco use—fast facts*, https://www.cdc.gov/tobacco/data_statistics/fact_sheets/fast_facts/index.htm, Accessed: 2021-02-04, 2020.
- [86] *Tobacco*, <https://www.who.int/news-room/fact-sheets/detail/tobacco>, Accessed: 2021-02-04, 2020.
- [87] A. Florescu, R. Ferrence, T. Einarson, P. Selby, O. Soldin, and G. Koren, “Methods for quantification of exposure to cigarette smoking and environmental tobacco smoke: Focus on developmental toxicology,” *Therapeutic drug monitoring*, vol. 31, no. 1, p. 14, 2009.
- [88] T. A. Burling, M. L. Stitzer, G. E. Bigelow, and A. M. Mead, “Smoking topography and carbon monoxide levels in smokers,” *Addictive Behaviors*, vol. 10, no. 3, pp. 319–323, Jan. 1985.
- [89] *Smoking topography study 2018*, <https://clinicaltrials.gov/ct2/show/NCT03498053>, Accessed: 2021-02-04, 2020.
- [90] A. A. Strasser, W. B. Pickworth, F. Patterson, and C. Lerman, “Smoking topography predicts abstinence following treatment with nicotine replacement therapy,” *Cancer Epidemiology and Prevention Biomarkers*, vol. 13, no. 11, pp. 1800–1804, 2004.
- [91] E. Lee, J. Potts, A. Waters, E. Moolchan, and W. Pickworth, “Smoking topography: Reliability and validity in dependent smokers,” *Nicotine & tobacco research : official journal of the Society for Research on Nicotine and Tobacco*, vol. 5, pp. 673–9, Oct. 2003.
- [92] M. C. Brinkman et al., “Comparison of true and smoothed puff profile replication on smoking behavior and mainstream smoke emissions,” *Chemical Research in Toxicology*, vol. 28, no. 2, pp. 182–190, 2015, PMID: 25536227. eprint: <https://doi.org/10.1021/tx500318h>.

- [93] V. Senyurek, M. Imtiaz, P. Belsare, S. Tiffany, and E. Sazonov, "Cigarette smoking detection with an inertial sensor and a smart lighter," *Sensors*, vol. 19, no. 3, 2019.
- [94] M. Imtiaz, D. Hossain, V. Senyurek, P. Belsare, S. Tiffany, and E. Sazonov, "Wearable egocentric camera as a monitoring tool of free-living cigarette smoking: A feasibility study," *Nicotine & tobacco research : official journal of the Society for Research on Nicotine and Tobacco*, vol. 22, Nov. 2019.
- [95] R. Alharbi, T. Stump, N. Vafaie, A. Pfammatter, B. Spring, and N. Alshurafa, "I can't be myself: Effects of wearable cameras on the capture of authentic behavior in the wild," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, Sep. 2018.
- [96] R. Alharbi, C. Feng, S. Sen, J. Jain, J. Hester, and N. Alshurafa, "Heatsight: Wearable low-power omni thermal sensing," in *2021 International Symposium on Wearable Computers*, 2021, pp. 108–112.
- [97] D. J. Roche, S. Bujarski, E. Hartwell, R. Green, and L. A. Ray, "Combined varenicline and naltrexone treatment reduces smoking topography intensity in heavy-drinking smokers," *Pharmacology Biochemistry and Behavior*, vol. 134, pp. 92–98, 2015.
- [98] R. I. Herning, R. T. Jones, J. Bachman, and A. H. Mines, "Puff volume increases when low-nicotine cigarettes are smoked.," *Br Med J (Clin Res Ed)*, vol. 283, no. 6285, pp. 187–189, 1981.
- [99] M. Mercincavage, J. L. Karelitz, C. L. Kreider, V. Souprountchouk, B. Albelda, and A. A. Strasser, "Comparing video observation to electronic topography device as a method for measuring cigarette puffing behavior," *Drug and Alcohol Dependence*, vol. 221, p. 108 623, 2021.
- [100] *Cress pocket*, <https://www.borgwaldt.com/en/products/smoking-vaping-machines/smoking-topography-devices/cress-pocket.html>, Accessed: 2021-02-04, 2021.
- [101] R. Robinson, E. Hensel, P. Morabito, and K. Roundtree, "Electronic cigarette topography in the natural environment," *PloS one*, vol. 10, no. 6, e0129296, 2015.
- [102] K. C. Ross and L. M. Juliano, "Smoking through a topography device diminishes some of the acute rewarding effects of smoking," *Nicotine & Tobacco Research*, vol. 18, no. 5, pp. 564–571, 2016.

- [103] M. H. Imtiaz, R. I. Ramos-Garcia, S. Wattal, S. Tiffany, and E. Sazonov, "Wearable sensors for monitoring of cigarette smoking in free-living: A systematic review," *Sensors*, vol. 19, no. 21, p. 4678, 2019.
- [104] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis, "Risq: Recognizing smoking gestures with inertial sensors on a wristband," in *Proceedings of the annual international conference on Mobile systems, applications, and services (Mobisys)*, 2014.
- [105] A. L. Skinner, C. J. Stone, H. Doughty, and M. R. Munafò, "Stopwatch: The preliminary evaluation of a smartwatch-based system for passive detection of cigarette smoking," *Nicotine and Tobacco Research*, vol. 21, no. 2, pp. 257–261, 2019.
- [106] M. Shoaib, H. Scholten, P. J. Havinga, and O. D. Incel, "A hierarchical lazy smoking detection algorithm using smartwatch sensors," in *IEEE International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2016.
- [107] V. Y. Senyurek, M. H. Imtiaz, P. Belsare, S. Tiffany, and E. Sazonov, "Smoking detection based on regularity analysis of hand to mouth gestures," *Biomedical signal processing and control*, vol. 51, pp. 106–112, 2019.
- [108] E. Sazonov, P. Lopez-Meyer, and S. Tiffany, "A wearable sensor system for monitoring cigarette smoking," *Journal of studies on alcohol and drugs*, vol. 74, no. 6, pp. 956–964, 2013.
- [109] P. M. Scholl and K. Van Laerhoven, "Lessons learned from designing an instrumented lighter for assessing smoking status," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, 2017, pp. 1016–1021.
- [110] M. H. Imtiaz, R. I. Ramos-Garcia, V. Y. Senyurek, S. Tiffany, and E. Sazonov, "Development of a multisensory wearable system for monitoring cigarette smoking behavior in free-living conditions," *Electronics*, vol. 6, no. 4, p. 104, 2017.
- [111] V. Senyurek, M. Imtiaz, P. Belsare, S. Tiffany, and E. Sazonov, "Cigarette smoking detection with an inertial sensor and a smart lighter," *Sensors*, vol. 19, no. 3, p. 570, 2019.
- [112] M. H. Imtiaz, D. Hossain, V. Y. Senyurek, P. Belsare, S. Tiffany, and E. Sazonov, "Wearable Egocentric Camera as a Monitoring Tool of Free-Living Cigarette Smoking: A Feasibility Study," *Nicotine & Tobacco Research*, Nov. 2019.

- [113] R. Alharbi, T. Stump, N. Vafaie, A. Pfammatter, B. Spring, and N. Alshurafa, “I can’t be myself: Effects of wearable cameras on the capture of authentic behavior in the wild,” *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 2, no. 3, pp. 1–40, 2018.
- [114] S. Shahi et al., “Impacts of image obfuscation on fine-grained activity recognition in ego-centric video,” in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2022, pp. 341–346.
- [115] T. Kawashima et al., “Action recognition from extremely low-resolution thermal image sequence,” in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2017, pp. 1–6.
- [116] E. Larson et al., “Heatwave: Thermal imaging for surface user interaction,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2011, pp. 2565–2574.
- [117] Y. Abdelrahman, A. Sahami Shirazi, N. Henze, and A. Schmidt, “Investigation of material properties for thermal imaging-based interaction,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 2015, pp. 15–18.
- [118] Y. Abdelrahman, E. Velloso, T. Dingler, A. Schmidt, and F. Vetere, “Cognitive heat: Exploring the usage of thermal imaging to unobtrusively estimate cognitive load,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 33, 2017.
- [119] F. Akbar et al., “Email makes you sweat: Examining email interruptions and stress using thermal imaging,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ACM, 2019, p. 668.
- [120] Y. Cho, N. Bianchi-Berthouze, N. Marquardt, and S. J. Julier, “Deep thermal imaging: Proximate material type recognition in the wild through deep learning of spatial surface temperature patterns,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ACM, 2018, p. 2.
- [121] M. L. Mauriello, M. Saha, E. B. Brown, and J. E. Froehlich, “Exploring novice approaches to smartphone-based thermographic energy auditing: A field study,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ACM, 2017, pp. 1768–1780.

- [122] M. L. Mauriello, B. McNally, and J. E. Froehlich, “Thermporal: An easy-to-deploy temporal thermographic sensor system to support residential energy audits,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ACM, 2019, p. 113.
- [123] M. L. Mauriello, L. Norooz, and J. E. Froehlich, “Understanding the role of thermography in energy auditing: Current practices and the potential for automated solutions,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 2015, pp. 1993–2002.
- [124] S. Shahi, M. Pedram, G. Fernandes, and N. Alshurafa, “Smartact: Energy efficient and real-time hand-to-mouth gesture detection using wearable rgb-t,” in *2022 IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2022, pp. 1–4.
- [125] S. Han, R. Nandakumar, M. Philipose, A. Krishnamurthy, and D. Wetherall, “Glimpsedata: Towards continuous vision-based personal analytics,” in *Proceedings of the 2014 workshop on physical analytics*, ACM, 2014, pp. 31–36.
- [126] F. Hu, P. He, S. Xu, Y. Li, and C. Zhang, “Fingertrak: Continuous 3d hand pose tracking by deep learning hand silhouettes captured by miniature thermal cameras on wrist,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–24, 2020.
- [127] S. Sen, V. Subbaraju, A. Misra, R. K. Balan, and Y. Lee, “The case for smartwatch-based diet monitoring,” in *2015 IEEE international conference on pervasive computing and communication workshops (PerCom workshops)*, IEEE, 2015, pp. 585–590.
- [128] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, *Mobilenetv2: Inverted residuals and linear bottlenecks*, 2019. arXiv: 1801.04381 [cs.CV].
- [129] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, “The jester dataset: A large-scale video dataset of human gestures,” in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2874–2882.
- [130] M. Chu et al., “Respiration rate and volume measurements using wearable strain sensors,” *NPJ digital medicine*, vol. 2, no. 1, pp. 1–9, 2019.
- [131] G. J. Fernandes et al., “Habitsense: A privacy-aware, ai-enhanced multimodal wearable platform for mhealth applications,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 8, no. 3, pp. 1–48, 2024.

- [132] C. Gurrin, A. F. Smeaton, A. R. Doherty, et al., “Lifelogging: Personal big data,” *Foundations and Trends® in information retrieval*, vol. 8, no. 1, pp. 1–125, 2014.
- [133] T. H. C. Nguyen, J.-C. Nebel, G. Hunter, and F. Florez-Revuelta, “Automated detection of hands and objects in egocentric videos, for ambient assisted living applications,” in *2018 14th International Conference on Intelligent Environments (IE)*, IEEE, 2018, pp. 91–94.
- [134] L. Gemming, J. Utter, and C. N. Mhurchu, “Image-assisted dietary assessment: A systematic review of the evidence,” *Journal of the Academy of Nutrition and Dietetics*, vol. 115, no. 1, pp. 64–77, 2015.
- [135] G. Welk, “Physical activity assessments for health-related research,” *Human Kinetics*, 2002.
- [136] R. Maddison et al., “Usefulness of wearable cameras as a tool to enhance chronic disease self-management: Scoping review,” *JMIR mHealth and uHealth*, vol. 7, no. 1, e10371, 2019.
- [137] A. Ksibi, A. S. D. Alluhaidan, A. Salhi, and S. A. El-Rahman, “Overview of lifelogging: Current challenges and advances,” *IEEE Access*, vol. 9, pp. 62 630–62 641, 2021.
- [138] S. Hodges et al., “Sensecam: A retrospective memory aid,” in *UbiComp 2006: Ubiquitous Computing: 8th International Conference, UbiComp 2006 Orange County, CA, USA, September 17-21, 2006 Proceedings 8*, Springer, 2006, pp. 177–193.
- [139] S. Shahi, M. Pedram, G. Fernandes, and N. Alshurafa, “Smartact: Energy efficient and real-time hand-to-mouth gesture detection using wearable rgb-t,” in *2022 IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, IEEE, 2022, pp. 1–4.
- [140] Z. Wu, C. Xiong, C.-Y. Ma, R. Socher, and L. S. Davis, “Adaframe: Adaptive frame selection for fast video recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1278–1287.
- [141] Y. Meng et al., “Ar-net: Adaptive frame resolution for efficient action recognition,” in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, Springer, 2020, pp. 86–104.
- [142] J. Lin, H. Duan, K. Chen, D. Lin, and L. Wang, “Ocsampler: Compressing videos to one clip with single-step sampling,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 894–13 903.

- [143] S. Alfasly et al., “Fastpicker: Adaptive independent two-stage video-to-video summarization for efficient action recognition,” *Neurocomputing*, vol. 516, pp. 231–244, 2023.
- [144] E. Niforatos, V. Lim, C. Vuerich, M. Langheinrich, and A. Bexheti, “Pulsecam: Biophysically driven life logging,” in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI ’15, New York, NY, USA: Association for Computing Machinery, 2015, pp. 1002–1009.
- [145] S. Fang, K. Mayer-Patel, and S. Nirjon, “Zencam: Context-driven control of autonomous body cameras,” in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, IEEE, 2019, pp. 41–48.
- [146] A. Ghodrati, B. E. Bejnordi, and A. Habibian, “Frameexit: Conditional early exiting for efficient video recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 608–15 618.
- [147] M. A. Hassan and E. Sazonov, “Selective content removal for egocentric wearable camera in nutritional studies,” *IEEE Access*, vol. 8, pp. 198 615–198 623, 2020.
- [148] R. Possas, S. P. Caceres, and F. Ramos, “Egocentric activity recognition on a budget,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5967–5976.
- [149] Y. Zhi, Z. Tong, L. Wang, and G. Wu, “Mgsampler: An explainable sampling strategy for video action recognition,” in *Proceedings of the IEEE/CVF International conference on Computer Vision*, 2021, pp. 1513–1522.
- [150] S. Tan, T. Nagarajan, and K. Grauman, “Egodistill: Egocentric head motion distillation for efficient video understanding,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 33 485–33 498, 2023.
- [151] B. Korbar, D. Tran, and L. Torresani, “Scsampler: Sampling salient clips from video for efficient action recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6232–6242.
- [152] R. Alharbi et al., “Smokemon: Unobtrusive extraction of smoking topography using wearable energy-efficient thermal,” *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 6, no. 4, pp. 1–25, 2023.

- [153] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, “Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1949–1957.
- [154] R. Alharbi, S. Sen, A. Ng, N. Alshurafa, and J. Hester, “Actisight: Wearer foreground extraction using a practical rgb-thermal wearable,” in *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, IEEE, 2022, pp. 237–246.
- [155] T. Ishihara, K. M. Kitani, W.-C. Ma, H. Takagi, and C. Asakawa, “Recognizing hand-object interactions in wearable camera videos,” in *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 1349–1353.
- [156] S. Michibata, K. Inoue, M. Yoshioka, and A. Hashimoto, “Cooking activity recognition in egocentric videos with a hand mask image branch in the multi-stream cnn,” in *Proceedings of the 12th Workshop on Multimedia for Cooking and Eating Activities*, 2020, pp. 1–6.
- [157] M. He et al., “Inertial strengthened clip model for zero-shot multimodal egocentric activity recognition,” in *2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2024, pp. 1–6.
- [158] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, “Real-time hand tracking under occlusion from an egocentric rgb-d sensor,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1154–1163.
- [159] T. Ohkawa, K. He, F. Sener, T. Hodan, L. Tran, and C. Keskin, “Assemblyhands: Towards egocentric activity understanding via 3d hand pose estimation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 12 999–13 008.
- [160] L. Zhang, S. Zhou, S. Stent, and J. Shi, “Fine-grained egocentric hand-object segmentation: Dataset, model, and applications,” in *European Conference on Computer Vision*, Springer, 2022, pp. 127–145.
- [161] C. Zhong, A. R. Reibman, H. M. Cordoba, and A. J. Deering, “Hand-hygiene activity recognition in egocentric video,” in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2019, pp. 1–6.
- [162] V.-D. Le et al., “Hand activity recognition from automatic estimated egocentric skeletons combining slow fast and graphical neural networks,” *Vietnam Journal of Computer Science*, vol. 10, no. 01, pp. 75–100, 2023.

- [163] B. Pei et al., “Egovideo: Exploring egocentric foundation model and downstream adaptation,” *arXiv preprint arXiv:2406.18070*, 2024.
- [164] B. Xu, Z. Wang, Y. Du, Z. Song, S. Zheng, and Q. Jin, “Egonce++: Do egocentric video-language models really understand hand-object interactions?” *arXiv preprint arXiv:2405.17719*, 2024.
- [165] S. Bansal, M. Wray, and D. Damen, “Hoi-ref: Hand-object interaction referral in egocentric vision,” *arXiv preprint arXiv:2404.09933*, 2024.
- [166] J. Xu et al., “Retrieval-augmented egocentric video captioning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 525–13 536.
- [167] Meta Platforms, Inc., *Battery life on ray-ban meta ai glasses*, <https://www.meta.com/help/ai-glasses/303057485648146/>, Accessed: 2025-04-28, 2025.
- [168] G. Harmon, *Ray-ban meta smart glasses get an on-device assistant, with battery life limitations*, <https://www.emarketer.com/content/ray-ban-meta-smart-glasses-on-device-assistant--with-battery-life-limitations>, Accessed: 2025-04-28, 2024.
- [169] R. Carter, *Snap spectacles 5 review: The latest snap ar glasses*, <https://www.xrtoday.com/augmented-reality/snap-spectacles-5-review-the-latest-snap-ar-glasses/>, Accessed: 2025-04-28, 2024.
- [170] F. Ding, Y. Zhu, X. Wen, G. Liu, and C. X. Lu, “Thermohands: A benchmark for 3d hand pose estimation from egocentric thermal images,” *arXiv preprint arXiv:2403.09871*, 2024.
- [171] Y. Ye et al., “Affordance diffusion: Synthesizing hand-object interactions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 479–22 489.
- [172] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordon, P. Labatut, and D. Novotny, “Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 901–10 911.

- [173] H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 2847–2854.
- [174] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*, PmLR, 2020, pp. 1597–1607.
- [175] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [176] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.
- [177] E. Hoffer and N. Ailon, “Deep metric learning using triplet network,” in *Similarity-based pattern recognition: third international workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, Springer, 2015, pp. 84–92.
- [178] S. Naderiparizi, P. Zhang, M. Philipose, B. Priyantha, J. Liu, and D. Ganesan, “Glimpse: A programmable early-discard camera architecture for continuous mobile vision,” in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, 2017, pp. 292–305.
- [179] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?” In *ICML*, vol. 2, 2021, p. 4.
- [180] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, “Swinir: Image restoration using swin transformer,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1833–1844.
- [181] P. Wang et al., “Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution,” *arXiv preprint arXiv:2409.12191*, 2024.
- [182] E. J. Hu et al., “Lora: Low-rank adaptation of large language models,” *ICLR*, vol. 1, no. 2, p. 3, 2022.

- [183] D. Shan, J. Geng, M. Shu, and D. F. Fouhey, “Understanding human hands in contact at internet scale,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9869–9878.
- [184] F. Zhang et al., “Mediapipe hands: On-device real-time hand tracking,” *arXiv preprint arXiv:2006.10214*, 2020.
- [185] K. Higuchi, R. Yonetani, and Y. Sato, “Egoscanning: Quickly scanning first-person videos with egocentric elastic timelines,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 6536–6546.
- [186] J. Engel et al., “Project aria: A new tool for egocentric multi-modal ai research,” *arXiv preprint arXiv:2308.13561*, 2023.
- [187] Y. Zhang, X. Li, and I. Marsic, “Multi-label activity recognition using activity-specific features and activity correlations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 625–14 635.
- [188] S. S. Rautaray and A. Agrawal, “Vision based hand gesture recognition for human computer interaction: A survey,” *Artificial intelligence review*, vol. 43, pp. 1–54, 2015.
- [189] B. K. Chakraborty, D. Sarma, M. K. Bhuyan, and K. F. MacDorman, “Review of constraints on vision-based gesture recognition for human–computer interaction,” *IET Computer Vision*, vol. 12, no. 1, pp. 3–15, 2018.
- [190] N. Gillian and J. A. Paradiso, “The gesture recognition toolkit,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3483–3487, 2014.
- [191] J. Gong, X.-D. Yang, and P. Irani, “Wristwhirl: One-handed continuous smartwatch input using wrist gestures,” in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 861–872.
- [192] Y. Iravantchi, M. Goel, and C. Harrison, “Beamband: Hand gesture sensing with ultrasonic beamforming,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–10.
- [193] J. Lien et al., “Soli: Ubiquitous gesture sensing with millimeter wave radar,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–19, 2016.

- [194] E. Wu, Y. Yuan, H.-S. Yeo, A. Quigley, H. Koike, and K. M. Kitani, “Back-hand-pose: 3d hand pose estimation for a wrist-worn camera via dorsum deformation network,” in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020, pp. 1147–1160.
- [195] R. Aigner et al., “Understanding mid-air hand gestures: A study of human preferences in usage of gesture types for hci,” *Microsoft Research TechReport MSR-TR-2012-111*, vol. 2, no. MSR-TR-2012-111, p. 30, 2012.
- [196] Y. Lou, W. Wu, R.-D. Vatavu, and W.-T. Tsai, “Personalized gesture interactions for cyber-physical smart-home environments,” *Science China Information Sciences*, vol. 60, no. 7, pp. 1–15, 2017.
- [197] U. Oh and L. Findlater, “The challenges and potential of end-user gesture customization,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2013, pp. 1129–1138.
- [198] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, “User-defined gestures for surface computing,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2009, pp. 1083–1092.
- [199] M. A. Nacenta, Y. Kamber, Y. Qiang, and P. O. Kristensson, “Memorability of pre-designed and user-defined gesture sets,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2013, pp. 1099–1108.
- [200] H. Dong, A. Danesh, N. Figueroa, and A. El Saddik, “An elicitation study on gesture preferences and memorability toward a practical hand-gesture vocabulary for smart televisions,” *IEEE access*, vol. 3, pp. 543–555, 2015.
- [201] T. Ouyang and Y. Li, “Bootstrapping personal gesture shortcuts with the wisdom of the crowd and handwriting recognition,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 2895–2904.
- [202] H. Xia, M. Glueck, M. Annett, M. Wang, and D. Wigdor, “Iteratively designing gesture vocabularies: A survey and analysis of best practices in the hci literature,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 29, no. 4, pp. 1–54, 2022.
- [203] L. Anthony, Y. Kim, and L. Findlater, “Analyzing user-generated youtube videos to understand touchscreen use by people with motor impairments,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2013, pp. 1223–1232.

- [204] L.-B. Bilius, O.-C. Ungurean, and R.-D. Vatavu, “Understanding wheelchair users’ preferences for on-body, in-air, and on-wheelchair gestures,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–16.
- [205] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, “A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities,” *ACM Computing Surveys*, 2023.
- [206] A. Parnami and M. Lee, “Learning from few examples: A summary of approaches to few-shot learning,” *arXiv preprint arXiv:2203.04291*, 2022.
- [207] J. Wan, Q. Ruan, W. Li, and S. Deng, “One-shot learning gesture recognition from rgb-d data using bag of features,” *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2549–2582, 2013.
- [208] Z. Lu, S. Qin, L. Li, D. Zhang, K. Xu, and Z. Hu, “One-shot learning hand gesture recognition based on lightweight 3d convolutional neural networks for portable applications on mobile systems,” *IEEE Access*, vol. 7, pp. 131 732–131 748, 2019.
- [209] L. Li, S. Qin, Z. Lu, K. Xu, and Z. Hu, “One-shot learning gesture recognition based on joint training of 3d resnet and memory module,” *Multimedia Tools and Applications*, vol. 79, pp. 6727–6757, 2020.
- [210] X. Xu et al., “Enabling hand gesture customization on wrist-worn devices,” in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–19.
- [211] Y. Zou and L. Cheng, “A transfer learning model for gesture recognition based on the deep features extracted by cnn,” *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 5, pp. 447–458, 2021.
- [212] U. Côté-Allard et al., “Deep learning for electromyographic hand gesture signal classification using transfer learning,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 27, no. 4, pp. 760–771, 2019.
- [213] Q. Wen et al., “Time series data augmentation for deep learning: A survey,” *arXiv preprint arXiv:2002.12478*, 2020.
- [214] Z. Ji et al., “Survey of hallucination in natural language generation,” *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

- [215] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *2018 international interdisciplinary PhD workshop (IIPhDW)*, IEEE, 2018, pp. 117–122.
- [216] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, “Meta-learning in neural networks: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.
- [217] J. Vanschoren, “Meta-learning: A survey,” *arXiv preprint arXiv:1810.03548*, 2018.
- [218] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S. F. Atashzar, and A. Mohammadi, “Fs-hgr: Few-shot learning for hand gesture recognition via electromyography,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 29, pp. 1004–1015, 2021.
- [219] X. Zeng, C. Wu, and W.-B. Ye, “User-definable dynamic hand gesture recognition based on doppler radar and few-shot learning,” *IEEE Sensors Journal*, vol. 21, no. 20, pp. 23 224–23 233, 2021.
- [220] G. B. Mo, J. J. Dudley, and P. O. Kristensson, “Gesture knitter: A hand gesture design tool for head-mounted mixed reality applications,” in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–13.
- [221] Y. Zhou, C. Li, Z.-Q. Cheng, Y. Geng, X. Xie, and M. Keuper, “Hypergraph transformer for skeleton-based action recognition,” *arXiv preprint arXiv:2211.09590*, 2022.
- [222] A. Fathi, A. Farhadi, and J. M. Rehg, “Understanding egocentric activities,” in *2011 international conference on computer vision*, IEEE, 2011, pp. 407–414.
- [223] R. L. Klatzky, “Allocentric and egocentric spatial representations: Definitions, distinctions, and interconnections,” in *Spatial cognition: An interdisciplinary approach to representing and processing spatial knowledge*, Springer, 1998, pp. 1–17.
- [224] S. Canavan, W. Keyes, R. McCormick, J. Kunnumpurath, T. Hoelzel, and L. Yin, “Hand gesture recognition using a skeleton-based feature representation with a random regression forest,” in *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 2364–2368.
- [225] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, “Skeleton-based dynamic hand gesture recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 1–9.

- [226] M. Yassen and S. Jusoh, “A systematic review on hand gesture recognition techniques, challenges and applications,” *PeerJ Computer Science*, vol. 5, e218, 2019.
- [227] D. Kim et al., “Digits: Freehand 3d interactions anywhere using a wrist-worn gloveless sensor,” in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 2012, pp. 167–176.
- [228] C. Harrison, D. Tan, and D. Morris, “Skinput: Appropriating the body as an input surface,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2010, pp. 453–462.
- [229] A. Akl, C. Feng, and S. Valaee, “A novel accelerometer-based gesture recognition system,” *IEEE transactions on Signal Processing*, vol. 59, no. 12, pp. 6197–6205, 2011.
- [230] H. Wen, J. Ramos Rojas, and A. K. Dey, “Serendipity: Finger gesture recognition using an off-the-shelf smartwatch,” in *Proceedings of the 2016 CHI conference on human factors in computing systems*, 2016, pp. 3847–3851.
- [231] G. Laput, R. Xiao, and C. Harrison, “Viband: High-fidelity bio-acoustic sensing using commodity smartwatch accelerometers,” in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 321–333.
- [232] M. J. Cheok, Z. Omar, and M. H. Jaward, “A review of hand gesture and sign language recognition techniques,” *International Journal of Machine Learning and Cybernetics*, vol. 10, pp. 131–153, 2019.
- [233] A. Kapitanov, A. Makhlyarchuk, and K. Kvanchiani, “Hagrid-hand gesture recognition image dataset,” *arXiv preprint arXiv:2206.08219*, 2022.
- [234] F. Yang, Y. Wu, S. Sakti, and S. Nakamura, “Make skeleton-based action recognition model smaller, faster and better,” in *Proceedings of the 1st ACM International Conference on Multimedia in Asia*, 2019, pp. 1–6.
- [235] P. Wang, W. Li, S. Liu, Z. Gao, C. Tang, and P. Ogunbona, “Large-scale isolated gesture recognition using convolutional neural networks,” in *2016 23rd international conference on pattern recognition (ICPR)*, IEEE, 2016, pp. 7–12.
- [236] H.-I. Lin, M.-H. Hsu, and W.-K. Chen, “Human hand gesture recognition using a convolution neural network,” in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, IEEE, 2014, pp. 1038–1043.

- [237] A. Mohanty, S. S. Rambhatla, and R. R. Sahay, “Deep gesture: Static hand gesture recognition using cnn,” in *Proceedings of International Conference on Computer Vision and Image Processing: CVIP 2016, Volume 2*, Springer, 2017, pp. 449–461.
- [238] B. Doosti, “Hand pose estimation: A survey,” *arXiv preprint arXiv:1903.01013*, 2019.
- [239] J. C. Nunez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Velez, “Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition,” *Pattern Recognition*, vol. 76, pp. 80–94, 2018.
- [240] J. Hou, G. Wang, X. Chen, J.-H. Xue, R. Zhu, and H. Yang, “Spatial-temporal attention res-
tcn for skeleton-based dynamic hand gesture recognition,” in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.
- [241] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, Jun. 2017, pp. 1126–1135.
- [242] D. Wu, F. Zhu, and L. Shao, “One shot learning gesture recognition from rgb-d images,” in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, 2012, pp. 7–12.
- [243] J. Wan, G. Guo, and S. Z. Li, “Explore efficient local features from rgb-d data for one-shot learning gesture recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 8, pp. 1626–1639, 2015.
- [244] J. Konečný and M. Hagara, “One-shot-learning gesture recognition using hog-hof features,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2513–2532, 2014.
- [245] H. J. Escalante, I. Guyon, V. Athitsos, P. Jangyodsuk, and J. Wan, “Principal motion components for one-shot gesture recognition,” *Pattern Analysis and Applications*, vol. 20, pp. 167–182, 2017.
- [246] L. Zhang et al., “Bomw: Bag of manifold words for one-shot learning gesture recognition from kinect,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2562–2573, 2017.

- [247] K. Stewart, G. Orchard, S. B. Shrestha, and E. Neftci, “Online few-shot gesture learning on a neuromorphic processor,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 4, pp. 512–521, 2020.
- [248] L. Li, S. Qin, Z. Lu, D. Zhang, K. Xu, and Z. Hu, “Real-time one-shot learning gesture recognition based on lightweight 3d inception-resnet with separable convolutions,” *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1173–1192, 2021.
- [249] H. Dang, L. Mecke, F. Lehmann, S. Goller, and D. Buschek, *How to prompt? opportunities and challenges of zero- and few-shot learning for human-ai interaction in creative applications of generative models*, 2022. arXiv: 2209.01390 [cs.HC].
- [250] S. Park, S. D. Mello, P. Molchanov, U. Iqbal, O. Hilliges, and J. Kautz, “Few-shot adaptive gaze estimation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.
- [251] C. Zhao, J. Wang, L. Li, X. Qu, and J. Xiao, *Adaptive few-shot learning algorithm for rare sound event detection*, 2022. arXiv: 2205.11738 [cs.SD].
- [252] V. D. Lai, T. H. Nguyen, and F. Dernoncourt, “Extensively matching for few-shot learning event detection,” in *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, C. Bonial et al., Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 38–45.
- [253] B. Kulis et al., “Metric learning: A survey,” *Foundations and Trends® in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2013.
- [254] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, “Meta-transfer learning for few-shot learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 403–412.
- [255] L. Torrey and J. Shavlik, “Transfer learning in handbook of research on machine learning applications and trends: Algorithms,” *Methods, and Techniques* (ed. E. Olivas, J. Guerrero, M. Martinez-Sober, J. Magdalena-Benedito, & A. Serrano López), pp. 242–264,
- [256] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. Van Der Maaten, “Simpleshot: Revisiting nearest-neighbor classification for few-shot learning,” *arXiv preprint arXiv:1911.04623*, 2019.

- [257] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, “Rethinking few-shot image classification: A good embedding is all you need?” In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, Springer, 2020, pp. 266–282.
- [258] X. Ying, “An overview of overfitting and its solutions,” in *Journal of physics: Conference series*, IOP Publishing, vol. 1168, 2019, p. 022 022.
- [259] P. Hu, C. Tang, K. Yin, and X. Zhang, “Wigr: A practical wi-fi-based gesture recognition system with a lightweight few-shot network,” *Applied Sciences*, vol. 11, no. 8, p. 3329, 2021.
- [260] J. Rajendran, A. Irpan, and E. Jang, “Meta-learning requires meta-augmentation,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 5705–5715, 2020.
- [261] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, “The jester dataset: A large-scale video dataset of human gestures,” in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019, pp. 0–0.
- [262] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and S. Z. Li, “Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 56–64.
- [263] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, “Skeleton-based dynamic hand gesture recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2016.
- [264] E. Choi, H. Kim, and M. K. Chung, “A taxonomy and notation method for three-dimensional hand gestures,” *International Journal of Industrial Ergonomics*, vol. 44, no. 1, pp. 171–188, 2014.
- [265] Z. Lv, A. Halawani, S. Feng, S. Ur Réhman, and H. Li, “Touch-less interactive augmented reality game on vision-based wearable device,” *Personal and Ubiquitous Computing*, vol. 19, pp. 551–567, 2015.
- [266] M. Mazouzi, P. Alavesa, T. Koskela, and T. Ojala, “Ghost hunters: Ambient light and hand gesture utilization in mobile augmented reality games,” in *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, 2016, pp. 365–367.

- [267] J. J. Dudley, H. Schuff, and P. O. Kristensson, “Bare-handed 3d drawing in augmented reality,” in *Proceedings of the 2018 designing interactive systems conference*, 2018, pp. 241–252.
- [268] H. Francke, J. Ruiz-del-Solar, and R. Verschae, “Real-time hand gesture detection and recognition using boosted classifiers and active learning,” in *Advances in Image and Video Technology: Second Pacific Rim Symposium, PSIVT 2007 Santiago, Chile, December 17-19, 2007 Proceedings 2*, Springer, 2007, pp. 533–547.
- [269] D. Wu et al., “Deep dynamic neural networks for multimodal gesture segmentation and recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 8, pp. 1583–1597, 2016.
- [270] G. Zhu, L. Zhang, P. Shen, J. Song, S. A. A. Shah, and M. Bennamoun, “Continuous gesture segmentation and recognition using 3dcnn and convolutional lstm,” *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1011–1021, 2018.
- [271] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, “A unified framework for gesture recognition and spatiotemporal gesture segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 9, pp. 1685–1699, 2008.
- [272] A. Antoniou, H. Edwards, and A. Storkey, “How to train your maml,” *arXiv preprint arXiv:1810.09502*, 2018.